SAXOTECH Online
Tags Manual

1. Tags Manual . . . 12
1.1 Chapter 1 Introduction . . . 12
1.2 Chapter 2 Publicus ini . . . 16
1.2.1 ADCLASSES section . . . 16
1.2.2 ADX section . . . 18
1.2.3 AGENDACLASSSUBJECTMAPPING section . . . 19
1.2.4 AGENDA section . . . 19
1.2.5 ALIASREWRITE section . . . 20
1.2.6 ARTSUBCLASS section . . . 20
1.2.7 CACHE section . . . 20
1.2.8 CAPTCHA section . . . 24
1.2.9 CCE section . . . 25
1.2.10 CLASSIFIEDSIMAGE section . . . 26
1.2.11 CLICKSHAREPRODUCTS section . . . 26
1.2.12 CLICKSHARE section . . . 26
1.2.13 COLLECTMOSTREADSTATS section . . . 27
1.2.14 CONTENTVALIDATION section . . . 28
1.2.15 DEBUG section . . . 28
1.2.16 EDITINTERFACE section . . . 28
1.2.17 EVENTSDEF section . . . 30
1.2.18 EXTLOGIN section . . . 31
1.2.19 EXTRAFIELDDEF section . . . 32
1.2.20 EXTRAMENUCPEDITORIAL1 section . . . 33
1.2.21 FACEBOOK section . . . 33
1.2.22 FASTSEARCH section . . . 33
1.2.23 FORUM section . . . 34
1.2.24 FRONTPAGESUBCLASS section . . . 35
1.2.25 GENERAL section . . . 35
1.2.26 GOOGLEMAPS section . . . 51
1.2.27 HOSTSITECODES section . . . 52
1.2.28 IMAGE section . . . 52
1.2.29 LIGHTUSERSTORYLIST . . . 53
1.2.30 MODULES section . . . 53
1.2.31 MULTIPRIORITY section . . . 57
1.2.32 MULTITABSTORYEDIT Section . . . 57
1.2.33 NETGUEST section . . . 57
1.2.34 NEWSLETTER section . . . 57
1.2.35 OBJECTRANKING section . . . 57
1.2.36 OMNITURE_FRONTPAGE_MAPPING section . . . 58
1.2.37 OMNITURE section . . . 58
1.2.38 OVRSUBCLASS section . . . 59
1.2.39 POLL section . . . 59
1.2.40 PRIOROTATEGROUPS section . . . 59
1.2.41 PUBLISHTIMES section . . . 60
1.2.42 QA section . . . 60
1.2.43 SEARCHSTRINGDEFS section . . . 60
1.2.44 SECTIONSUBCLASS section . . . 61
1.2.45 SINGLETABSTORYEDIT section . . . 61
1.2.46 StoryEdit section . . . 62
1.2.47 TEMPLATEMAP section . . . 65
1.2.48 URLValidation section . . . 65
1.2.49 UserSubmitCCE Section . . . 65
1.2.50 UserSubmit section . . . 66
1.2.51 UserVerify section . . . 68
1.2.52 Web server ini file . . . 68
1.3 Chapter 3 PBCSdb.ini . . . 69
1.3.1 Database connection pool . . . 69
1.3.2 Database servers . . . 70
1.3.3 Edit template settings . . . 70
1.3.4 General cache settings . . . 71
1.3.5 GLOBAL section . . . 72
1.3.6 Language file caching . . . 72
1.3.7 Online Search . . . 72
1.3.8 Plugins . . . 72
1.3.9 SC_Map . . . 73
1.3.10 Template settings . . . 74
1.3.11 Thread pool . . . 75
1.4 Chapter 4 HTML and Includes . . . 75
1.4.1 Document structure . . . 76
1.4.2 Fonts . . . 76
1.4.3 Meta tags . . . 79
1.4.3.1 Article90.pbo . . . 79
1.4.3.2 ArticleHead90.pbo . . . 79
1.4.3.3 ArticleHead.pbo meta tag . . . 79
1.4.4 Pbs filecheck . . . 79
1.4.5 Pbs getcontent . . . 80
1.4.6 Pbs include . . . 81
1.4.7 Server side includes . . . 82

1.4.8 Special graphics directory . . . 83
1.5 Chapter 5 Templates . . . 84
1.5.1 Export templates . . . 84
1.5.1.1 article_xml_asset.pbo . . . 84
1.5.1.2 article_xml_connection.pbo . . . 84
1.5.1.3 article_xml_paragraph.pbo . . . 85
1.5.1.4 article_xml.pbo . . . 85
1.5.1.5 templates_Sms_xml . . . 87
1.5.2 Folder structure . . . 88
1.5.3 Palm templates . . . 88
1.5.4 Setting MIME Types . . . 88
1.5.5 Special templates . . . 89
1.5.6 Template mapping . . . 89
1.5.7 Template names and URLs . . . 90
1.5.8 WAP templates . . . 92
1.5.9 Zones . . . 92
1.6 Chapter 6 Object Files . . . 93
1.6.1 About Object Files . . . 93
1.6.2 Article classes . . . 98
1.6.3 Global function . . . 100
1.6.4 Global variables . . . 100
1.6.5 Inline objects . . . 101
1.6.6 Localdata.ini . . . 102
1.6.7 Object script . . . 102
1.6.8 Odd and Even variable picture handling . . . 115
1.6.9 Pbs macro . . . 117
1.6.9.1 Election2008.pbo . . . 119
1.6.10 Position-based design . . . 119
1.6.11 System variables . . . 119
1.7 Chapter 7 Menu Tags . . . 120
1.7.1 pbs categorytree . . . 121
1.7.1.1 CategoryTree.pbo . . . 121
1.7.1.2 CategoryTreeItemActive.pbo . . . 121
1.7.1.3 CategoryTreeItemPassive.pbo . . . 122
1.7.1.4 CategoryTreeItemSeparatorColl.pbo . . . 122
1.7.1.5 CategoryTreeItemSeparatorExp.pbo . . . 122
1.7.1.6 CategoryTreeItemSeparatorNone.pbo . . . 122
1.7.2 pbs ft . . . 122
1.7.3 pbs leftmenu . . . 123
1.7.4 pbs menu-image . . . 124
1.7.5 pbs menu-text . . . 124
1.7.6 pbs profiletree . . . 124
1.7.6.1 ProfileTree.pbo . . . 124
1.7.6.2 ProfileTreeItem.pbo . . . 125
1.7.6.3 ProfileTreeItemActive.pbo . . . 125
1.7.6.4 ProfileTreeItemPassive.pbo . . . 125
1.7.6.5 ProfileTreeItemSeparatorColl.pbo . . . 125
1.7.6.6 ProfileTreeItemSeparatorExp.pbo . . . 125
1.7.7 pbs qwiknavsearch . . . 125
1.7.7.1 QwikNavSearchForm.pbo . . . 125
1.8 Chapter 8 Section Front Tags . . . 126
1.8.1 pbs oversiktnotiser . . . 126
1.8.2 pbs sectionfront . . . 126
1.9 Chapter 9 Article Tags . . . 129
1.9.1 ArticleConnectionItem.pbo . . . 130
1.9.2 pbs articlefield . . . 130
1.9.3 pbs articledate . . . 130
1.9.4 pbs artikkel-urls . . . 130
1.9.5 pbs articlemultimedia . . . 130
1.9.6 pbs photogallery-links . . . 131
1.9.7 pbs keywordlinks . . . 131
1.9.7.1 KeywordLinkItem.pbo . . . 131
1.9.7.2 KeywordLinks.pbo . . . 131
1.9.8 pbs themelinks . . . 131
1.9.8.1 ThemeLinkItem.pbo . . . 132
1.9.8.2 ThemeLinks.pbo . . . 132
1.9.9 Other links . . . 132
1.9.10 pbs article-links . . . 132
1.9.11 pbs article . . . 133
1.9.11.1 Article.pbo . . . 141
1.9.11.2 ArticleByline.pbo . . . 144
1.9.11.3 ArticleConnections.pbo . . . 146
1.9.11.4 ArticleExtraPictureItem.pbo . . . 146
1.9.11.5 ArticleExtraPictures.pbo . . . 146
1.9.11.6 ArticleFacts.pbo . . . 146
1.9.11.7 ArticleForumConnectionItem.pbo . . . 147
1.9.11.8 ArticleForumConnections.pbo . . . 147
1.9.11.9 ArticleGalleryConnectionItem.pbo . . . 147
1.9.11.10 ArticleGalleryConnections.pbo . . . 147
1.9.11.11 ArticleHead.pbo . . . 147

1.9.11.12 ArticleLinkItem.pbo . . . 148
1.9.11.13 ArticleLinks.pbo . . . 148
1.9.11.14 ArticleLocation.pbo . . . 148
1.9.11.15 ArticleModified.pbo . . . 149
1.9.11.16 ArticleMultiMedia.pbo . . . 149
1.9.11.17 ArticleMultiMediaExtension.pbo . . . 149
1.9.11.18 ArticleMultimediaLinkItem.pbo . . . 150
1.9.11.19 ArticleMultiMediaList.pbo . . . 150
1.9.11.20 ArticleMultiMediaListItem.pbo . . . 150
1.9.11.21 ArticleMultimediaListItemExtension.pbo . . . 150
1.9.11.22 ArticleMultimediaListItemXXX.pbo . . . 150
1.9.11.23 ArticleParagraph.pbo . . . 151
1.9.11.24 ArticlePhotoGalleryThumbnailItem.pbo . . . 151
1.9.11.25 ArticlePhotoGalleryThumbnails.pbo . . . 151
1.9.11.26 ArticlePicture.pbo . . . 151
1.9.11.27 ArticlePublished.pbo . . . 153
1.9.11.28 ArticleSummary.pbo . . . 153
1.9.11.29 InvalidArticle.pbo . . . 153
1.9.11.30 Tips.pbo and articles . . . 153
1.9.11.31 WAPArticle.pbo . . . 153
1.9.11.32 WAPArticleContent.pbo . . . 154
1.9.11.33 WAPArticleParagraph.pbo . . . 154
1.9.12 ArticleGeoDataItem.pbo . . . 154
1.9.13 ArticleGeoDataItems.pbo . . . 154
1.10 Chapter 10 Overview Tags . . . 154
1.10.1 pbs agent . . . 155
1.10.1.1 Agent.pbo . . . 159
1.10.1.2 AgentItem.pbo . . . 159
1.10.2 pbs agentcategory . . . 160
1.10.3 pbs history . . . 160
1.10.3.1 History.pbo . . . 161
1.10.3.2 HistoryItem.pbo . . . 161
1.10.4 pbs keywordindex . . . 161
1.10.4.1 KeywordIndexHeader.pbo . . . 162
1.10.4.2 KeywordIndexLink.pbo . . . 162
1.10.4.3 KeywordIndexTitle.pbo . . . 162
1.10.5 pbs mostread . . . 162
1.10.5.1 MostRead.pbo . . . 164
1.10.5.2 MostReadItem.pbo . . . 164
1.10.6 pbs newslist . . . 164
1.10.6.1 News.pbo . . . 173
1.10.6.2 NewsItem.pbo . . . 174
1.10.6.3 NewsItemArticleConnectionItem.pbo . . . 178
1.10.6.4 NewsItemConnectionItem.pbo . . . 178
1.10.6.5 NewsItemExternalLinkItem.pbo . . . 178
1.10.6.6 NewsItemExternalLinks.pbo . . . 178
1.10.6.7 NewsItemForumConnectionItem.pbo . . . 178
1.10.6.8 NewsItemForumGroupLink.pbo . . . 179
1.10.6.9 NewsItemGalleryConnectionItem.pbo . . . 179
1.10.6.10 NewsItemInternalLinks.pbo . . . 179
1.10.6.11 NewsItemKeywordLinks.pbo . . . 179
1.10.6.12 NewsItemKeywordLinksItem.pbo . . . 179
1.10.6.13 NewsItemMultiMediaExtension.pbo . . . 179
1.10.6.14 NewsItemThemeLinkItem.pbo . . . 179
1.10.6.15 NewsItemThemeLinks.pbo . . . 180
1.10.6.16 RelStoriesItem.pbo . . . 180
1.10.6.17 RelStoriesItems.pbo . . . 180
1.10.6.18 URL Parameters and overview tags . . . 180
1.10.7 pbs related . . . 180
1.10.7.1 ThemeLinkItem.pbo and overview tags . . . 181
1.10.7.2 ThemeLinks.pbo and pbs related . . . 181
1.10.8 pbs subcategories . . . 181
1.10.8.1 SubcatSimpleLink.pbo . . . 182
1.11 Chapter 11 Searching . . . 182
1.11.1 Guided search . . . 183
1.11.1.1 pbs searchresult_guided . . . 183
1.11.1.1.1 ImageSearchNavigator.pbo . . . 184
1.11.1.1.2 ImageSearchNavigatorItem.pbo . . . 184
1.11.1.1.3 SearchNavigator.pbo . . . 184
1.11.1.1.4 SearchNavigatorItem.pbo . . . 184
1.11.1.1.5 StorySearchNavigator.pbo . . . 184
1.11.1.1.6 StorySearchNavigatorItem.pbo . . . 184
1.11.2 Image search . . . 185
1.11.2.1 pbs imagesearchresult . . . 186
1.11.2.1.1 SearchResultImage.pbo . . . 186
1.11.2.1.2 SearchResultImageItem.pbo . . . 186
1.11.2.1.3 SearchResultImageRow.pbo . . . 187
1.11.3 Result page . . . 187
1.11.3.1 Building the page without object files . . . 187
1.11.3.1.1 pbs search_criteria . . . 187

1.11.3.1.2 pbssearch_hits . . . 187
1.11.3.1.3 pbs search_next_link . . . 187
1.11.3.1.4 pbs search_previous_link . . . 187
1.11.3.1.5 pbs search_totalhits . . . 188
1.11.3.1.6 pbs searchresult without object files . . . 188
1.11.3.2 pbs searchresult . . . 188
1.11.3.2.1 SearchError.pbo . . . 189
1.11.3.2.2 SearchResult.pbo . . . 189
1.11.3.2.3 SearchResultItem.pbo . . . 190
1.11.4 Search form . . . 190
1.11.5 Search other websites . . . 193
1.11.6 Search syntax . . . 194
1.12 Chapter 12 Classifieds . . . 196
1.12.1 Ad inserts . . . 196
1.12.1.1 Ad inserts using pbs classifieds . . . 197
1.12.1.2 Ads in stories . . . 197
1.12.1.3 As a banner ad . . . 197
1.12.1.4 Importing ad inserts . . . 198
1.12.2 Classifieds data retrieval tags . . . 199
1.12.3 pbs classifieddetails . . . 200
1.12.3.1 ClassifiedDetails.pbo . . . 200
1.12.3.2 ClassifiedPrintDetails.pbo . . . 201
1.12.3.3 ClassifiedsRelated.pbo . . . 201
1.12.4 pbs classifiedsfields . . . 201
1.12.4.1 ClassifiedFieldsXXX.pbo . . . 202
1.12.4.2 ClassifiedFieldsXXXItem.pbo . . . 202
1.12.4.3 ClassifiedFieldsXXXRow.pbo . . . 202
1.12.5 Classifieds pbs LastUpdate . . . 203
1.12.6 Classifieds search form . . . 203
1.12.7 Classifieds setup . . . 204
1.12.8 pbs classifieds . . . 204
1.12.8.1 Classifieds.pbo . . . 209
1.12.8.2 ClassifiedsGroupHeader.pbo . . . 210
1.12.8.3 ClassifiedsItem.pbo . . . 211
1.13 Chapter 13 Custom Content Engine . . . 211
1.13.1 CCE data retrieval tags . . . 212
1.13.2 CCE pbs LastUpdate . . . 212
1.13.3 CCE search form . . . 213
1.13.4 CCE setup . . . 214
1.13.5 CCE tip a friend . . . 215
1.13.6 pbs cce . . . 215
1.13.6.1 CCE_ModuleShortName.pbo . . . 224
1.13.6.2 CCE_ModuleShortNameGroupHeader.pbo . . . 225
1.13.6.3 CCE_ModuleShortNameItem.pbo . . . 225
1.13.6.4 cce_ModuleShortNameNavigator.pbo . . . 227
1.13.6.5 cce_ModuleShortNameNavigatorItem.pbo . . . 227
1.13.6.6 CCE_ModuleShortNamePageLink . . . 228
1.13.6.7 CCE_ModuleShortNamePageLinkItem . . . 228
1.13.6.8 CCE_MPLACE.pbo . . . 229
1.13.6.9 CCE_MPLACEitem.pbo . . . 229
1.13.6.10 CCE-search_ovr.pbs . . . 229
1.13.7 pbs ccedetails . . . 229
1.13.7.1 CCE_ModuleShortNameDetails.pbo . . . 230
1.13.7.2 CCE_ModuleShortNamePrintDetail.pbo . . . 230
1.13.7.3 CCE_ModuleShortNameRelated.pbo . . . 230
1.13.8 pbs ccefields . . . 230
1.13.8.1 CCE_ModuleShortNameFieldsFdNmRw . . . 233
1.13.8.2 CCE_ModuleShortNameFieldsFieldName.pbo . . . 233
1.13.8.3 CCE_ModuleShortNameFieldsFieldNameItem.pbo . . . 233
1.13.9 pbs usersubmitform . . . 233
1.13.10 SubmitCCEObject action . . . 234
1.14 Chapter 14 Banner Ads . . . 235
1.14.1 pbs ad . . . 236
1.14.1.1 AdSpot.pbo . . . 236
1.14.1.2 FlashBanner.pbo . . . 237
1.14.1.3 LayerAdSpot.pbo . . . 237
1.14.2 SAXOTECH Online ads on other sites . . . 237
1.15 Chapter 15 Links . . . 238
1.15.1 Links.pbo . . . 238
1.15.2 pbs links . . . 238
1.15.3 pbs linkscategorylist . . . 239
1.15.3.1 LinksItem.pbo . . . 240
1.15.3.2 LinksSubcat.pbo . . . 241
1.15.3.3 Links vote form . . . 241
1.16 Chapter 16 White Pages . . . 242
1.16.1 pbs ptemailsearch . . . 242
1.17 Chapter 17 Yellow Pages . . . 243
1.17.1 pbs ypcategorylist . . . 243
1.17.1.1 DirectoryCategoryList.pbo . . . 243
1.17.1.2 DirectoryCategoryListItem.pbo . . . 243

1.17.2 pbs ypdetails . . . 244
1.17.2.1 DirectoryDetails.pbo . . . 244
1.17.2.2 DirectoryDetailsItem.pbo . . . 244
1.17.3 pbs ypsubcategorylist . . . 245
1.17.3.1 DirectorySubcategoryList.pbo . . . 245
1.17.3.2 DirectorySubcategoryListItem.pbo . . . 245
1.18 Chapter 18 Guides . . . 245
1.18.1 Guides free text search . . . 246
1.18.1.1 SearchResultGuide.pbo . . . 246
1.18.1.2 SearchResultGuideItem.pbo . . . 246
1.18.2 pbs ptguides . . . 247
1.18.2.1 GuideSubcatItem.pbo . . . 247
1.18.3 pbs ptguidesearchitem . . . 248
1.18.3.1 GuideSearchItem.pbo . . . 248
1.18.3.2 GuideSearchItemExtraItem.pbo . . . 249
1.18.4 pbs ptguidesearchlist . . . 249
1.18.4.1 GuideBusinessHours.pbo . . . 250
1.18.4.2 GuideSearchList.pbo . . . 250
1.18.4.3 GuideSearchListHeader.pbo . . . 252
1.18.4.4 GuideSearchListItem.pbo . . . 252
1.18.5 pbs ptguidesearchoptions . . . 252
1.18.5.1 GuideSearchOptionsItem.pbo . . . 254
1.18.5.2 GuideSearchOptionsRow.pbo . . . 254
1.18.6 Guide search form . . . 254
1.19 Chapter 19 Q&A . . . 256
1.19.1 pbs pollarchive . . . 257
1.19.1.1 PollArchiveItem.pbo . . . 257
1.19.2 pbs polldata . . . 257
1.19.2.1 PollDataResults.pbo . . . 259
1.19.2.2 PollDataResultsItem.pbo . . . 259
1.19.2.3 PollDataResultsItemEntry.pbo . . . 259
1.19.2.4 PollForm.pbo . . . 260
1.19.2.5 PollFormAlternatives.pbo . . . 260
1.19.2.6 PollFormAlternativesItem.pbo . . . 260
1.19.2.7 PollFormItem.pbo . . . 260
1.19.2.8 Polls_Error.pbo with pbs polldata . . . 260
1.19.2.9 Polls_Saved.pbo with pbs polldata . . . 260
1.19.2.10 Poll Statistics . . . 260
1.19.3 pbs pollmini . . . 261
1.19.3.1 Pollmini.pbo . . . 262
1.19.3.2 PollMiniChart.pbo . . . 262
1.19.3.3 PollMiniNoPoll.pbo . . . 262
1.19.4 pbs quest_multipage . . . 262
1.19.4.1 MultiPageQuestIndex.pbo . . . 262
1.19.4.2 MultiPageQuestPage.pbo . . . 263
1.19.4.3 MultiPageQuestPageX.pbo . . . 263
1.19.4.4 MultiPageQuestSubmit.pbo . . . 263
1.19.4.5 MultiPageQuestWrongAnswer.pbo . . . 264
1.19.4.6 Multipage self test . . . 264
1.19.4.7 MultiQuestAltItem.pbo . . . 264
1.19.4.8 Poll_Error.pbo with pbs quest_multipage . . . 265
1.19.4.9 Poll_Saved.pbo with pbs quest_multipage . . . 265
1.19.5 pbs questform . . . 265
1.19.5.1 Polls_Error.pbo with pbs questform . . . 265
1.19.5.2 Polls_Saved.pbo with pbs questform . . . 265
1.19.5.3 QuestForm.pbo . . . 265
1.19.5.4 QuestFormAlternativesItem.pbo . . . 265
1.19.5.5 QuestFormItem.pbo . . . 266
1.20 Chapter 20 Events . . . 266
1.20.1 Printer friendly calendar . . . 267
1.20.2 pbs events_previous_link . . . 267
1.20.3 Events setup . . . 267
1.20.4 Events search form . . . 268
1.20.5 Add events form . . . 272
1.20.6 pbs eventlist . . . 275
1.20.6.1 EventDateHeader.pbo . . . 279
1.20.6.2 EventDetails.pbo . . . 279
1.20.6.3 EventItem.pbo . . . 279
1.20.6.4 EventLocationHeader.pbo . . . 280
1.20.6.5 EventSubjectHeader.pbo . . . 280
1.20.7 pbs events_next_link . . . 280
1.20.8 pbs cinemalist . . . 281
1.20.8.1 MovieList.pbo . . . 281
1.20.8.2 MovieListItem.pbo . . . 281
1.21 Chapter 21 Forums . . . 282
1.21.1 Message filtering . . . 283
1.21.2 Message form . . . 283
1.21.2.1 ForumNewMessage.pbo . . . 283
1.21.2.2 ForumNewMessageExtLogin.pbo . . . 285
1.21.2.3 ForumNewMessageUnavailable.pbo . . . 285

1.21.3 pbs Captcha . . . 285
1.21.3.1 Captcha.pbo . . . 286
1.21.4 pbs forumoptions . . . 286
1.21.4.1 ForumPreferences.pbo . . . 287
1.21.4.2 ForumPreferencesSaved.pbo . . . 288
1.21.4.3 ForumPrivateEmail.pbo . . . 288
1.21.4.4 ForumPrivateEmailMessage.pbo . . . 289
1.21.4.5 ForumPrivateEmailSent.pbo . . . 289
1.21.5 pbs news_content . . . 290
1.21.5.1 ForumMessage.pbo . . . 291
1.21.5.2 ForumMessageAttachmentItem.pbo . . . 293
1.21.5.3 ForumMessageThreadItem.pbo . . . 293
1.21.5.4 ForumThemeItem.pbo . . . 294
1.21.5.5 ForumThemes.pbo . . . 295
1.21.5.6 ForumThemesGroupItem.pbo . . . 295
1.21.6 pbs news_count . . . 295
1.21.7 pbs news_grouplink . . . 295
1.21.7.1 ForumGroupLink.pbo . . . 296
1.21.8 pbs news_groups . . . 296
1.21.8.1 ForumGroupHeadlineItem.pbo . . . 296
1.21.8.2 ForumGroupItem.pbo . . . 297
1.21.8.3 ForumGroups.pbo . . . 297
1.21.9 pbs news_hotthreads . . . 298
1.21.9.1 ForumHotThreadItem.pbo . . . 299
1.21.9.2 ForumHotThreads.pbo . . . 299
1.21.10 pbs news_latest . . . 299
1.21.10.1 ForumLatestMessageItem.pbo . . . 300
1.21.10.2 ForumLatestMessages.pbo . . . 301
1.21.11 pbs news_messages . . . 301
1.21.11.1 ForumMessageItem.pbo . . . 302
1.21.11.2 ForumMessages.pbo . . . 303
1.21.11.3 ForumMessagesMessageThreadItem.pbo . . . 304
1.21.11.4 ForumNavBar.pbo . . . 306
1.21.11.5 ForumNavBarItem.pbo . . . 306
1.21.12 pbs search . . . 306
1.21.13 pbs searchresult and forums . . . 308
1.21.13.1 SearchResultForum.pbo . . . 308
1.21.13.2 SearchResultForumItem.pbo . . . 308
1.21.14 pbs userverify . . . 309
1.21.14.1 UserVerificationMail.pbo . . . 309
1.21.14.2 UserVerify.pbo . . . 309
1.22 Chapter 22 Net Guest . . . 310
1.22.1 NetGuest.pbo . . . 310
1.22.2 pbs netguest . . . 311
1.22.2.1 NetGuestQuestionItem.pbo . . . 312
1.22.3 pbs netguest_list . . . 312
1.22.3.1 NetGuestList.pbo . . . 313
1.22.3.2 NetGuestListItem.pbo . . . 313
1.22.4 pbs netguest_questionform . . . 313
1.22.4.1 NetGuestQuestionForm.pbo . . . 313
1.23 Chapter 23 User Generated Content . . . 314
1.23.1 pbs userarticleform . . . 316
1.23.1.1 ArticleForm.pbo . . . 317
1.23.1.2 ArticleFormMediaItem.pbo . . . 318
1.23.1.3 ArticleFormSectionItem.pbo . . . 319
1.23.1.4 UserSubArticles_ArticleFormNotLoggedIn . . . 319
1.23.2 pbs usersubmitform for user generated content . . . 319
1.23.2.1 UserSubmitForm.pbo . . . 321
1.23.2.2 UserSubmitFormArticleMediaItem.pbo . . . 322
1.23.2.3 UserSubmitFormArticleSectionItem.pbo . . . 323
1.23.2.4 UserSubmitFormCategoryList.pbo . . . 323
1.23.2.4.1 UserSubmitFormCategoryListItem.pbo . . . 323
1.23.2.5 UserSubmitFormCCEFieldList.pbo . . . 323
1.23.2.5.1 UserSubmitFormCCEFieldListItem.pbo . . . 324
1.23.2.6 UserSubmitFormCCEModule.pbo . . . 324
1.23.2.7 UserSubmitFormCCERecurrencePattern.pbo . . . 324
1.23.2.8 UserSubmitFormErrorFailed.pbo . . . 327
1.23.2.9 UserSubmitFormErrorInvalidInput.pbo . . . 327
1.23.2.10 UserSubmitFormErrorNotCreator.pbo . . . 328
1.23.2.11 UserSubmitFormErrorNotLoggedIn.pbo . . . 328
1.23.2.12 UserSubmitFormExtraFields.pbo . . . 328
1.23.2.12.1 UserSubmitFormExtraFieldsItem.pbo . . . 328
1.23.2.13 UserSubmitFormPhotoGalleryPhotoItem.pbo . . . 329
1.23.2.14 UserSubmitFormTaxonomyList.pbo . . . 329
1.23.2.14.1 UserSubmitFormTaxonomyListItem.pbo . . . 329
1.23.3 UserSubmitFormArticle.pbo . . . 329
1.23.4 UserSubmitFormArticleCategoryList.pbo . . . 329
1.23.5 UserSubmitFormArticleCategoryListItem.pbo . . . 329
1.23.6 UserSubmitFormArticleErrorFailed.pbo . . . 330
1.23.7 UserSubmitFormArticleErrorInvalidInput.pbo . . . 330

1.23.8 UserSubmitFormArticleErrorNotCreator.pbo . . . 330
1.23.9 UserSubmitFormArticleErrorNotLoggedIn.pbo . . . 330
1.23.10 UserSubmitFormArticleExtraFields.pbo . . . 330
1.23.11 UserSubmitFormArticleExtraFieldsItem.pbo . . . 330
1.23.12 UserSubmitFormArticleTaxonomyList.pbo . . . 330
1.23.13 UserSubmitFormArticleTaxonomyListItem.pbo . . . 330
1.23.14 UserSubmitFormCCE.pbo . . . 330
1.23.15 UserSubmitFormCCECategoryList.pbo . . . 330
1.23.16 UserSubmitFormCCECategoryListItem.pbo . . . 330
1.23.17 UserSubmitFormCCEErrorFailed.pbo . . . 330
1.23.18 UserSubmitFormCCEErrorInvalidInput.pbo . . . 330
1.23.19 UserSubmitFormCCEErrorNotCreator.pbo . . . 331
1.23.20 UserSubmitFormCCEErrorNotLoggedIn.pbo . . . 331
1.23.21 UserSubmitFormCCERecurrencePattern.pbon . . . 331
1.23.22 UserSubmitFormCCETaxonomyList.pbo . . . 331
1.23.23 UserSubmitFormCCETaxonomyListItem.pbom . . . 331
1.23.24 UserSubmitFormPhotoGallery.pbo . . . 331
1.23.25 UserSubmitFormPhotoGalleryCategoryList.pbo . . . 331
1.23.26 UserSubmitFormPhotoGalleryCategoryListItem.pbo . . . 331
1.23.27 UserSubmitFormPhotoGalleryErrorFailed.pbo . . . 331
1.23.28 UserSubmitFormPhotoGalleryErrorInvalidInput.pbo . . . 331
1.23.29 UserSubmitFormPhotoGalleryErrorNotCreator.pbo . . . 331
1.23.30 UserSubmitFormPhotoGalleryErrorNotLoggedIn.pbo . . . 331
1.23.31 UserSubmitFormPhotoGalleryExtraFields.pbo . . . 331
1.23.32 UserSubmitFormPhotoGalleryExtraFieldsItem.pbo . . . 331
1.23.33 UserSubmitFormPhotoGalleryTaxonomyList.pbo . . . 332
1.23.34 UserSubmitFormPhotoGalleryTaxonomyListItem.pbo . . . 332
1.23.35 User Submitted Stories . . . 332
1.24 Chapter 24 Subscriptions . . . 332
1.24.1 Login page . . . 333
1.24.1.1 pbs login . . . 333
1.24.1.2 pbs logininfo . . . 333
1.24.1.3 pbs target . . . 333
1.24.2 pbs changepassword . . . 333
1.24.3 pbs global . . . 335
1.24.4 pbs subscriberdata . . . 335
1.24.4.1 SubscriberData.pbo . . . 336
1.24.5 pbs userid . . . 336
1.24.6 Self-created users . . . 336
1.25 Chapter 25 User Registration . . . 338
1.25.1 Advanced login . . . 338
1.25.1.1 pbs ureg_email . . . 340
1.25.1.2 pbs ureg_message . . . 340
1.25.2 Edit user registration . . . 340
1.25.3 Log out . . . 341
1.25.4 Lost password . . . 341
1.25.4.1 PwdSent.pbo . . . 342
1.25.5 pbs adregusertargets . . . 343
1.25.6 pbs extra_income . . . 343
1.25.7 pbs reguserinfo . . . 344
1.25.7.1 Userreginfo.pbo . . . 344
1.25.8 Setup user registration . . . 345
1.26 Chapter 26 Staff Listing . . . 349
1.26.1 pbs departmentlist . . . 350
1.26.1.1 DepList.pbo . . . 350
1.26.1.2 DepListItem.pbo . . . 350
1.26.2 pbs departmentoverview . . . 350
1.26.2.1 DepOverview.pbo . . . 351
1.26.2.2 DepOverviewGroupHeader.pbo . . . 351
1.26.2.3 DepOverviewHeader.pbo . . . 351
1.26.2.4 DepOverviewItem.pbo . . . 351
1.26.3 pbs staffinfo . . . 352
1.26.3.1 StaffInfo.pbo . . . 352
1.27 Chapter 27 Tip Functions . . . 353
1.27.1 External tip page . . . 353
1.27.2 pbs article-tiplink . . . 354
1.27.2.1 ArtTips.pbo . . . 354
1.27.2.2 ArtTipsText.pbo . . . 355
1.27.2.3 ArtTipsTextHtml.pbo . . . 355
1.27.2.4 ClassTipsText.pbo . . . 355
1.27.2.5 ClassTipsTextHTML.pbo . . . 355
1.27.2.6 GuideTipsText.pbo . . . 356
1.27.2.7 GuideTipsTextHTML.pbo . . . 356
1.27.2.8 Tips.pbo . . . 356
1.27.2.9 TipsText.pbo . . . 356
1.27.3 Setup tip functions . . . 357
1.28 Chapter 28 Images . . . 357
1.28.1 Access control for images . . . 357
1.28.1.1 PhotoGalleryExtLogin.pbo . . . 357
1.28.2 Story-linked photo gallery . . . 358

1.28.3 Generate graphical text . . . 358
1.28.4 Image settings . . . 360
1.28.5 Order picture . . . 362
1.28.6 pbs link and images . . . 364
1.28.7 pbs photogallery . . . 365
1.28.7.1 ArticleConnectionItem.pbo and images . . . 365
1.28.7.2 ArticleConnections.pbo and images . . . 365
1.28.7.3 PhotoGallery.pbo . . . 366
1.28.8 pbs photogallery-latest . . . 367
1.28.8.1 PhotoGalleryLatest.pbo . . . 369
1.28.8.2 PhotoGalleryLatestItem.pbo . . . 369
1.28.9 pbs photogallery-thumbnails . . . 370
1.28.9.1 PhotoGalleryThumbnailItem.pbo . . . 371
1.28.9.2 PhotoGalleryThumbnails.pbo . . . 372
1.28.10 pbs ptshowimage . . . 372
1.28.10.1 ShowImage.pbo . . . 373
1.28.11 Photo album export . . . 373
1.28.11.1 ExpPhotoAlbum.pbo . . . 374
1.28.11.2 ExpPhotoAlbumItem.pbo . . . 375
1.28.11.3 ExpPhotoAlbums.pbo . . . 375
1.28.12 Set picture as desktop background . . . 375
1.28.13 Watermark images . . . 376
1.28.14 pbs ptimages . . . 377
1.28.14.1 ImageItem.pbo . . . 379
1.28.14.2 ImageItemRowEnd.pbo . . . 380
1.28.14.3 ImageItemRowStart.pbo . . . 380
1.28.15 PhotoGalleryItemExtraFieldItem.pbo . . . 380
1.28.16 PhotoGalleryItemExtraFields.pbo . . . 380
1.28.17 PhotoGalleryLatestItemExtraFieldItem.pbo . . . 380
1.28.18 PhotoGalleryLatestItemExtraFields.pbo . . . 380
1.28.19 PhotoGalleryLatestItemTaxonomy.pbo . . . 380
1.28.20 PhotoGalleryLatestTaxonomyItem.pbo . . . 381
1.28.21 PhotoGalleryTaxonomyItem.pbo . . . 381
1.29 Chapter 29 Newsletter . . . 381
1.29.1 Newsletter template . . . 381
1.29.1.1 Newsletter.txt . . . 381
1.29.1.2 NewsletterAd.txt . . . 382
1.29.1.3 NewsletterArticle.txt . . . 382
1.29.1.4 NewsletterCategory.txt . . . 382
1.29.1.5 NewsletterLinks.txt . . . 383
1.29.1.6 NewsletterOAMSAdHTML.pbo . . . 383
1.29.1.7 NewsletterOAMSAdSpot.txt . . . 383
1.29.1.8 NewsletterSniff.txt . . . 383
1.29.2 pbs Newslad . . . 384
1.29.3 pbs newsletter_subscriptionform . . . 384
1.29.3.1 NewsletterActivated.pbo . . . 384
1.29.3.2 NewsLetterActivatedError.pbo . . . 384
1.29.3.3 NewsletterRemoved.pbo . . . 384
1.29.3.4 NewsLetterRemoveError.pbo . . . 384
1.29.3.5 NewsLetterRemoveMail.pbo . . . 385
1.29.3.6 NewsLetterSubscribeMail.pbo . . . 385
1.29.3.7 NewsLetterSubscriptionForm.pbo . . . 385
1.29.4 Setup newsletters . . . 386
1.29.5 Subscription page with Sniff Search . . . 387
1.29.6 Web based management for newsletters . . . 388
1.29.6.1 SubcatSimpleLink1.pbo . . . 390
1.30 Chapter 30 Newsletter Composer . . . 390
1.30.1 Main template . . . 391
1.30.2 NewsletterAd.pbo . . . 391
1.30.3 NewsletterArticle.pbo . . . 391
1.30.4 Newsletter email . . . 392
1.30.5 Web based management . . . 392
1.31 Chapter 31 Omniture . . . 392
1.31.1 pbs omniturecode . . . 393
1.31.1.1 URL action . . . 393
1.31.2 Setup Omniture . . . 393
1.32 Chapter 32 US Weather . . . 393
1.32.1 pbs accuwforecast . . . 393
1.32.1.1 AccuWForecast.pbo . . . 394
1.32.2 pbs accuwcurrent . . . 395
1.32.2.1 AccuWCurrent.pbo . . . 396
1.32.3 pbs accuwname . . . 396
1.32.4 pbs accuwmap . . . 396
1.32.4.1 AccuWMap.pbo . . . 397
1.32.5 pbs accuwoptions . . . 397
1.33 Chapter 33 European Weather . . . 398
1.33.1 pbs weatherforecast . . . 398
1.33.1.1 WeatherForecast.pbo . . . 399
1.33.1.2 WeatherForecastColItem.pbo . . . 399
1.33.1.3 WeatherForecastItem.pbo . . . 399

1.33.1.4 WeatherForecastRowItem.pbo . . . 399
1.33.2 pbs weathergraph . . . 399
1.33.2.1 WeatherGraph.pbo . . . 399
1.33.3 pbs weatherlastupdate . . . 400
1.33.4 pbs weatherobservations . . . 400
1.33.4.1 WeatherObservation.pbo . . . 400
1.33.4.2 WeatherObservationItem.pbo . . . 400
1.33.5 pbs weatherpicture . . . 400
1.33.5.1 WeatherPicture.pbo . . . 401
1.33.6 pbs weathertext . . . 401
1.33.7 pbs weathertidesunmoon . . . 401
1.33.7.1 WeatherTideSunMoon.pbo . . . 401
1.33.8 Storm weather service . . . 401
1.34 Chapter 34 Horoscopes . . . 402
1.34.1 pbs horoscope . . . 402
1.35 Chapter 35 TV Guide . . . 402
1.35.1 pbs tv_getoptions . . . 403
1.35.2 pbs tv_LineUpData . . . 403
1.35.2.1 TVLineUpSelect.pbo . . . 404
1.35.2.2 TVLineUpZipSelect.pbo . . . 404
1.35.3 pbs tv_programinfo . . . 404
1.35.3.1 TvPrograminfo.pbo . . . 404
1.35.4 pbs tv_schedule . . . 404
1.35.4.1 TVSchedule.pbo . . . 405
1.35.4.2 TVScheduleHeaderRow.pbo . . . 405
1.35.4.3 TVScheduleRow.pbo . . . 405
1.35.4.4 TVScheduleRowCell.pbo . . . 405
1.36 Chapter 36 Movie Guide . . . 406
1.36.1 Link to own reviews . . . 406
1.36.2 pbs mv_movieinfo . . . 406
1.36.2.1 MVMovieInfo.pbo . . . 407
1.36.2.2 MVMovieInfoImageItem.pbo . . . 407
1.36.2.3 MVMovieInfoImageRow.pbo . . . 408
1.36.2.4 MVMovieInfoReviewItem.pbo . . . 408
1.36.3 pbs mv_schedule . . . 408
1.36.3.1 MVSchedule.pbo . . . 409
1.36.3.2 MVScheduleChainFooter.pbo . . . 409
1.36.3.3 MVScheduleChainHeader.pbo . . . 409
1.36.3.4 MVScheduleMovieItem.pbo . . . 409
1.36.3.5 MVScheduleTheaterFooter.pbo . . . 409
1.36.3.6 MVScheduleTheaterHeader.pbo . . . 409
1.36.4 Search form for movie guide . . . 409
1.37 Chapter 37 Other Tags . . . 410
1.37.1 Internet Explorer functions . . . 411
1.37.2 pbs calendar . . . 411
1.37.2.1 Calendar.pbo . . . 412
1.37.2.2 CalendarDateItem.pbo . . . 412
1.37.2.3 CalendarWeekItem.pbo . . . 413
1.37.3 pbs category . . . 413
1.37.4 pbs categorydescription . . . 413
1.37.5 pbs categoryid . . . 413
1.37.6 pbs catselect . . . 414
1.37.7 pbs datecalc . . . 414
1.37.8 pbs datechoice . . . 415
1.37.9 pbs delimiter . . . 416
1.37.9.1 Delimiter.pbo . . . 416
1.37.10 pbs dingbats . . . 416
1.37.10.1 DingBats.pbo . . . 416
1.37.11 pbs frontpagearchive . . . 416
1.37.11.1 FPArchive.pbo . . . 417
1.37.11.2 FPArchiveItem.pbo . . . 417
1.37.12 pbs getcookie . . . 417
1.37.13 pbs getheaderinfo . . . 417
1.37.13.1 HeaderInfo.pbo . . . 417
1.37.14 pbs GetKeywords . . . 419
1.37.14.1 KeywordList.pbo . . . 420
1.37.15 pbs lastupdated . . . 420
1.37.16 pbs link . . . 420
1.37.17 pbs metarefresh . . . 421
1.37.18 pbs parvariables name="site" . . .
1.37.19 pbs PbcsRequestRandomA . . . 421
1.37.20 pbs PbcsRequestRandomB . . . 421
1.37.21 pbs profileid . . . 421
1.37.22 pbs profilename . . . 422
1.37.22.1 ProfileName.pbo . . . 422
1.37.23 pbs prog . . . 422
1.37.24 pbs query . . . 422
1.37.24.1 CustomQuery_queryname.pbo . . . 423
1.37.24.2 CustomQuery_querynameitem.pbo . . . 423
1.37.25 pbs queryparam . . . 424

1.37.26 pbs random . . . 424
1.37.27 pbs script . . . 424
1.37.28 pbs sessioninfo . . . 426
1.37.28.1 ArticleExtLogin.pbo . . . 426
1.37.29 pbs SetHeaderInfo . . . 426
1.37.30 pbs showdate . . . 427
1.37.31 pbs statuslog . . . 427
1.37.32 pbs UniquePageID . . . 427
1.37.33 Pop-up calendar . . . 427
1.37.34 sessioninfo with static cache files . . . 428
1.38 Chapter 38 Community Site . . . 429
1.38.1 Other tags and community site . . . 429
1.38.2 pbs article and community site . . . 429
1.38.3 pbs buildmenu . . . 429
1.38.3.1 MenuGraphicItem.pbo . . . 429
1.38.3.2 MenuTextItem.pbo . . . 429
1.38.4 pbs lightid . . . 430
1.38.5 pbs links on community site . . . 430
1.38.6 pbs sectionfront on community site . . . 430
1.38.7 Community site setup . . . 430
1.38.8 Community site templates . . . 430
1.39 Chapter 39 Xsendmail . . . 431
1.39.1 XSendMail.pbo . . . 431
1.39.2 XSendMail Form . . . 431
1.39.3 XSendMailHTML.pbo . . . 433
1.39.4 XSendMailHTMLItem.pbo . . . 433
1.39.5 XSendMailItem.pbo . . . 433
1.39.6 XSendMail setup . . . 434
1.40 Chapter 40 URL Rewrite . . . 435
1.40.1 New URLs . . . 435
1.40.2 Potential problems . . . 436
1.40.3 URL Parameters . . . 436
1.40.4 URL Rewrite setup . . . 436
1.40.5 Using URL Rewrite . . . 439
1.41 Chapter 41 Go.dll . . . 440
1.41.1 guide_GuideAdItem . . . 440
1.41.2 guide_GuideMainItem . . . 440

Tags Manual

Welcome to the SAXOTECH Online Tags Manual. This manual covers the technical aspects of setting up a publication with SAXOTECH Online. It is targeted toward people who make and manage SAXOTECH Online templates. This manual is to be used as a reference manual in conjunction with the Template Training documentation.

Chapter 1 Introduction

AboutOnline

AboutThisManual

Caching

DatabaseConfiguration

DateFormats

DirectoryStructure

StoryJumps

TaxonomyProfiles

Zoning

About SAXOTECH Online

SAXOTECH Online is a database and template-driven system. It retrieves information from a database based on queries made by templates. Your job in building a SAXOTECH Online site, is to design templates that will properly pull the information from the database and display it in the correct manner.

This documentation contains the information about the Publicus.ini file, template structure, SAXOTECH Online tags and object files.

About this manual

SAXOTECH Online version

The version information for SAXOTECH Online is shown by accessing the editing interface. This documentation reflects the most recent version of Online, as released from SAXOTECH Development. If you are running an older version of the system, some features mentioned will not be available.

Prerequisites
Before you start working with this manual, you should have a firm grasp of HTML, web servers, and other Internet technologies. It is designed for site designers and administrators and is not meant to be a training manual.

Caching

SAXOTECH Online is a database-driven system, but all content is not retrieved from the database each time the website is loaded. The information is cached on the web server.

Publicus.ini file settings

The general CACHE section for your website are stored in publicus.ini in the root folder for your site.

Tag settings

It is also possible to set the cache time directly in a tag. This is done for tags for which you want to have a more frequent update cycle, like a breaking news listing. To do this you must run the tag uncached and add the option **cachetime=X** to the tag. X sets the interval (in minutes) at which the the tag is cached.

Overriding cache settings

You can override the cache settings for a page by adding **&cachetime=X** to the link to the page. X is the interval (in minutes) at which the page should be cached.

By putting an exclamation mark ! in a tag after {{pbs:}}you indicate that the content of the tag should not be cached by the system. For example, using the exclamation on the weather tag like this:

```
<pbs:!accuwcurrent>
```

will cause the system to always pull the weather from the database.

This is done to keep information up to date on a page that does not change very often. For example, a story is usually not modified after it is put into the system. However, it might be active on the site long after it is published if it is linked to another story by a theme. Web guests will want to see current weather information, not weather information from the time the story was published interested in seeing the weather at the time the Story was published. The exclamation mark will insure that the weather stays current.

If this function is used on too many tags, the result can crash the server. It should only be used on tags that are cached in the database. If you are unsure about using this function, contact SAXOTECH.

Database configuration

SAXOTECH Online requires a complex database design to run. Configuration of the database is not a task for the customer alone. SAXOTECH will assist you in setting up the database if you run your system yourself. If your solution is hosted by us, we will handle the configuration.

Date formats and time zones

Date formats

Date formats are different from country to country. SAXOTECH Online allows you to format the systemÕs date output as needed. There are two major ways of formatting dates:

1. SAXOTECH Online tags that output dates often have an option called date format. If the tag supports this option, it is described with the tag.
2. You can set the format directly in the variable inside the object file. If the date field supports formatting, it is documented in the description of the date field.

The codes used for formatting dates are described in the section Formatting dates.

Time zones

If your website is in a different time zone than the web server, all times on your website will be wrong if additional steps are not taken to correct this. An easy way to test if this is the case is to put the tag {{<pbs: showdate format=hh:mm>}}on a page and uncache it. If the time is correct, your settings are correct. If the time is wrong, please contact your server administrator.

Please see Web server ini file for more information.

Directory structure

SAXOTECH Online requires several folders on the web server to run. The folders are placed in various locations on the server and some of them are common for all sites installed on the server. They contain programs and templates that cannot be modified by the customer. These folders are maintained by SAXOTECH.

Story jumps

SAXOTECH Online can handle story jumps in several different ways. Depending on how you want your site to function, you select which method to use.

From the front page

There are multiple ways to do story jumps from the front page to either the story or the section front for the category in which the story is placed.

Automatic jump with different text

For each story, you can write a teaser that features text and a picture. The teaser will be shown on the front page and sectionfront.

Automatic jump with same text

There are two ways to do this.

1. You can use the summary on both the front page and the story.
2. You can not use the summary at all. In this case, the first paragraph of the story will be used.

From sectionfronts

The usual way of doing story jumps from a section front to a story is to display the summary on the sectionfront. If the summary does not exist, SAXOTECH Online will use the first paragraph of the story. You can adjust how much text you would like to use. Please see pbs sectionfront.

In stories

To do story jumps between stories, you have multiple options. Please see Article Tags.

Taxonomy and profiles

Taxonomy is a method for ordering content that can be used when information can be classified in a hierarchical structure

In SAXOTECH Online, a taxonomy is made up of key words. Each keyword must be unique and appear only once.

A basic taxonomy for a sports website could look like this

- Sports
  - Ball sports
  - Football
    - NFL
    - AFL
    - XFL
  - Baseball
    - MLB
    - AAA
  - Tennis
  - Ski sports
  - Bike sports
- Geography
  - World
  - Central America
  - Europe
  - North America
    - Canada
    - United States
      - California
      - Florida
        - Miami
        - Tampa
      - Georgia
      - Illinois
      - Maryland
        - Baltimore
        - Annapolis
      - Texas
  - South America

If we follow the NFL branch deeper, the branch could be split into conferences, and conferences into divisions and divisions into teams. In SAXOTECH Online, taxonomies are used with profiles.

Profile description

Profiles are used for extracting information based on the taxonomy words added to a story. By tagging all stories with keywords as deep in the hierarchy as you can go, you get the most flexibility. This is because the stories will be extracted when you select a keyword higher in the hierarchy.

The profiles are predefined extractions based on taxonomy keywords. You can use multiple taxonomy keywords, and you can also exclude taxonomy keywords.

Example
There is a story in the system about the Baltimore Ravens winning the Super Bowl. The Ravens are a National Football League (NFL) club from Baltimore, Maryland. The Super Bowl was in Tampa, Florida.

The story is tagged with the following words from the taxonomy: NFL, Baltimore and Tampa. The story will be extracted from the system for all profiles that use the defined keywords directly, or keywords that are on the same branch higher in the hierarchy. Higher than NFL would be Football and Ball Sports; higher than Baltimore and Tampa would be Maryland, Florida, United States and North America.

Profiles and categories

You can tie a profile to a category. This allows you to use the category template for design, and the profile for selection of content. Please see the _Categories _chapter in the _SAXOTECH Online User Manual _for information about tying a profile to a category.

Profiles and zones
You can also tie a profile to a zone. This allows you to set up a multi-zone website. This is described in the next chapter.

Zoning

Zoning enables multiple versions of websites based on geographical locations. For example, local news has different content based on geographical location. The zoning functionality allows you to build one website for different zones or languages and different templates will be shown based on the language or zone chosen by the website guest.

Zone as language

You can use zoning for selecting different templates based on the zone the user is in. This allows you to use different languages for Zones. However, this system does not allow you to create multiple versions of a story in different languages.

Creating zone profiles

You must set up the Zones before you can use them. You can assign Zones to each category, and you can set fallback Zones in the ini file. You must first create a geographical taxonomy that you will use to build the Zones.

Example taxonomy
Big Co is building their worldwide website. They have products assigned to the regions in which they are sold, and need to build Zones so that the products appear in the correct Zones. They build their taxonomy like this:

- World
    - Europe
        - Norway
        - Denmark
        - France
    - North America
        - United States
        - Canada

Creating profiles

To create a profile that shows all products sold in Canada, you build a profile that shows all entries with North America or Canada as a zone. To create a profile that shows all products sold in the US, you build a profile that shows all entries with North America or the US as a zone.

Category setup

When you create categories, you must assign names for all Zones to the categories. This allows you to have different names for the categories based on the website guest's zone. You can also assign profiles to each zone the category has. Please see the *SAXOTECH Online User Manual* for more information.

Building templates

For selecting a zone, please see Zones.

To include files based on zone, please see Server side includes.

Tags that use profiles

<pbs:menu-text>	Output the name assigned to the zone for the category.
<pbs:newslist>	The Newslist tag supports selection of object files based on zone.

Object files

Selecting object files based on zone. Please see Zones and Selecting which Object Files to Use for more information.

Tagging content

When a story is created, a taxonomy word must be assigned to the story. This word is later used by the profile to extract the right content based on zone.

Images scaler

Our images scaler can handle these formats:

Bmp,Pcx,Jpeg,TIFF,PNG,TGA,PBM,PGM,PPM and Gif

Chapter 2 Publicus ini

Publicus.ini controls the operation of the website. Unless you are absolutely sure what the settings do, you should not change these settings.

The Publicus.ini file should not be edited without contacting SAXOTECH.

Please note that some of the field names and options are in Norwegian. We are in the process of translating all tags and options.

Placement
The ini file is placed in the root folder for the website. Each website's root folder is usually a sub folder of \www and has the same name as the website. This folder is only readable by the customer and the web server.

Sample
See Publicus.ini in the sample set. This file contains most of the known parameters.

- ADCLASSES section
- ADX section
- AGENDACLASSSUBJECTMAPPING section
- AGENDA section
- ALIASREWRITE section
- ARTSUBCLASS section
- CACHE section
- CAPTCHA section
- CCE section
- CLASSIFIEDSIMAGE section
- CLICKSHAREPRODUCTS section
- CLICKSHARE section
- COLLECTMOSTREADSTATS section
- CONTENTVALIDATION section
- DEBUG section
- EDITINTERFACE section
- EVENTSDEF section
- EXTLOGIN section
- EXTRAFIELDDEF section
- EXTRAMENUCPEDITORIAL1 section
- FACEBOOK section
- FASTSEARCH section
- FORUM section
- FRONTPAGESUBCLASS section
- GENERAL section
- GOOGLEMAPS section
- HOSTSITECODES section
- IMAGE section
- LIGHTUSERSTORYLIST
- MODULES section
- MULTIPRIORITY section
- MULTITABSTORYEDIT Section
- NETGUEST section
- NEWSLETTER section
- OBJECTRANKING section
- OMNITURE_FRONTPAGE_MAPPING section
- OMNITURE section
- OVRSUBCLASS section
- POLL section
- PRIOROTATEGROUPS section
- PUBLISHTIMES section
- QA section
- SEARCHSTRINGDEFS section
- SECTIONSUBCLASS section
- SINGLETABSTORYEDIT section
- StoryEdit section
- TEMPLATEMAP section
- URLValidation section
- UserSubmitCCE Section
- UserSubmit section
- UserVerify section
- Web server ini file

ADCLASSES section

In this section you define the ad classes for the Classifieds module. These classes can be used in the templates to show a single type of ad. The value for class is set when the ad is created or imported.The format of the ad classes section is as follows:

```
[ADCLASSES]
running number="class"
```

The value used here must match the value used when the ads were created. Example:

```
[ADCLASSES]
0=General
1=Houses
3=Cars
5=Jobs
9=Display ads
6=Obituaries
7=Boats
8=Boats Display Ads
10=Celebrations
```

Each of the classes must be set up in an

```
[ADCLASSDEF]
```

section in the following format:

```
[ADCLASSDEF]
FieldInternalName="field label text",width,height
```

Note that the running number in the

```
[ADCLASSDEF]
```

section must be a two digit number. These sections are used to define the ad classes in this section. These classes can be used in the templates to show a single type of ad. The value for class is set when the ad is created or imported.

FieldInternalName	**The name used when referencing the field**
Field Label Text	**The label used in the edit interface**
Height	**Given in lines**
Width	**Given in character columns**

```
[ADCLASSDEF00]
AdText="Text",40,15

[ADCLASSDEF01]
AdText="Text",40,15

[ADCLASSDEF03]
AdText="Text",40,15

[ADCLASSDEF05]
AdText="Text",40,15

[ADCLASSDEF07]
AdText="Text",40,15
```

```
ADCLASSDEF06
AdText="Text",40,15
NAME="Name",30,1
AGE="Age",4,1
BIRTHDAY="Birth date",8,1
DIED="Date of death",8,1
```

Additional fields are defined in the example above. The uppercase names can be used in the object files as variables. The information that goes into the different fields must be present in separate fields in the import file, or the import program must extract them.

The definition of fields requires that the import program be tuned to the capabilities of your Classified system. This part of the ini file should not be modified without consulting SAXOTECH.

The following names are reserved for use by the system:
Title, Date, Text and Keyword.

ADX section

Adinsert	The OAMS Ad insert requires that you have the setting
	`Adinsert=1`
	in your
	`[AdX]`
	section. The setting must be made in the Publicus.ini file used by the site on which you want to use the functionality. If the setting is not present, the standard ads insert function will be used. Please see bannerads_Pbsad for more information.

Domain	The OAMS display dll does not support being placed behind a front-end cache/load balancer that changes the request host name. Instead it simply uses the host name in the request hitting the dll. To use a different host name, use the following setting: `Domain=http://my.oams.domain.com` If this setting exists, all click and image url's will be prefixed with this domain and not the host name in the request. Otherwise, host name is used.

FlashWMode	By default, a browser puts a flash object (and other embedded plug in objects) in the topmost layer, so it sits on top of all other DIVs, regardless of the z-index set on those DIVs. This can cause problem, for example with Java-scripted drop-down menus that slip under the flash object instead of on top of it.OAMS.dll supports the parameter wmode=transparent in the embed/object tags, if the Publicus.ini file on the site running the OAMS.dll has the following setting:
	`[AdX]`
	`FlashWMode=transparent`
	If FlashWMode is specified, it will be added as the value of a wmode parameter. If it is not specified in the ini file, the wmode parameter will not be set.

LabelInClassifiedsLink	Determine whether the label ID will be included in the redirect URL generated by OAMS. Enter `LabelInClassifiedsLink=1` if the label ID should be included. The default is 1.
SpotInClassifiedsLink	Determine whether the spot ID will be included in the redirect URL generated by OAMS. Enter `SpotInClassifiedsLink=1` if the spot ID should be included. The default is 1.

ShowClicksInCustomerLogin	Controls whether or not customers login in will see number of clicks (and clickrate) as part of the reports they can pull. Enter `ShowClicksInCustomerLogin=1` Default is that they don't see it.

AGENDACLASSSUBJECTMAPPING section

This section is used for mapping the different classes in the system to the correct events type.

```
[AGENDACLASSSUBJECTMAPPING]
Agenda0=Subject3
Agenda1=Subject0
Agenda2=Subject2
```

Agenda0	General events.
Agenda1	Movie events.
Agenda2	People events.
Subject 3	The Events class

AGENDA section

Under the heading AGENDA, you define the properties for the agenda.

Guidelink	Optional. Enables links between a guide item and an event. When this option is used, you must also add the events field definition below. `Guidelink=1`
HideSub-Subjects	Optional. Disables sub-subjects. The option is set like this: `HideSub-Subjects=1`
Images	Optional. Enables images in the main listing. The option is set like this: `Images=1`
Sites	Optional. Determines the site code from which events should be retrieved. The option is set like this: `Sites=XX,XY` 85 sites or 255 characters is the maximum number of sites that can be used.

TextLimit	Optional. Determines whether or not there is a limit on the text length on an event. Enabling this option limits events to 255 characters. The option is set like this: `TextLimit=0` By default **TextLimit** is enabled.

ALIASREWRITE section

Aliases created in the database cannot include wildcards for performance reasons. However, you can perform a **regexp** rewrite of the URL before the alias is interpreted. The rewrite rules are placed in the ALIASREWRITE section.

This rewrite can be either partial (it will be translated as alias after the rewrite) or complete (the URL will be used as defined in the rewrite). The presence of the

```
<pbcs>
```

macro in the rewritten result is used to determine whether the rewrite is complete or not. If present, the macro will be expanded and the result used. If not, the result will be looked up in the database.

Complete rewrites are much faster than partial rewrites.

Examples:

```
[ALIASREWRITE]
/sports=/local
/cgi-bin/sect.*?\?Name\x3DLOCALNEWS(&|\Z)=/local
/cgi-bin/sect.*?\?Name\x3D(.+?)(&|\Z)=/section?Category=$1
/sections/(.*)=/section?category=$1
/business=/section?Category=business
```

The format is

```
regexp=replacement
```

The first example is a partial rewrite and the /local value will be translated as a normal alias. The second line is a complete rewrite.

- Notice that substitution is supported so subexpression matches can be used on the right hand side.
- Notice the use of \x3D instead of = (hex 3D) on the left hand side. This is due to the way parameters are read from the ini file (the first equal sign separates the name and the value portion of the pair).
  Processing of the rewrite rules stops as soon as a match is found, so the most frequently used rewrite rules should be put towards the top of the section.

ARTSUBCLASS section

The artsubclass section defines the subclasses that should be available for the story. It is set up the same way as the FRONTPAGESUBCLASS section.

```
[ARTSUBCLASS]
a=Breaking news
b=Major event
```

CACHE section

These settings control the caching on the website. The section has the heading CACHE.

Article_Cachetime	Optional. Sets the cache time (in minutes) for stories only. It is set like this: `Article_Cachetime=720`

CacheClassifieds	Optional. Turns caching of classifieds on or off. This option should only be used when debugging a website. `CacheClassifieds=0`
CacheTime	Sets the cache time for the whole website. The option determines the minimum number of minutes that the page should be cached on the web server. The option is set like this: `CacheTime=120` Saxotech recommends setting cachetime to 1440. CacheTime should not be set below 720.

DisableNoCache	When
	`DisableNoCache=1`
	is set,
	`&nocache=1`
	in the URL does not have any effect. This means that refreshing stories from the edit interface does not have any effect on the main site unless refresh from edit interface refers to a site other than the main site. For example:
	`Main site = www.mainsite.com (disable Nocache=1)`
	`Edit site = edit.mainsite.com (domain=http://www.mainsite2.com)`
	`Update site = www.mainsite2.com`
	(the cache directory of this site is the same as the main site).

DisableTooBusyFunction	Disables the Instance limit function. `DisableTooBusyFunction=1`
EnablePostProcessRandomNumbers	Set this to 1 to enable the `<pbs:#@PbcsRequestRandomA#>`, `<pbs:#@PbcsRequestRandomB#>`, and `<pbs:#@UniquePageID#>` tags for post processing using this setting. The default is 0. `EnablePostProcessRandomNumbers=0|1`
EnablePostProcessSessionInfo	The `<pbs:#@sessioninfo#>` tag and the `<pbs:#sessioninfo#>` tag will only be moved to post processing in static cache files if this is set to 1. The default is 0. `EnablePostProcessSessionInfo=0|1`

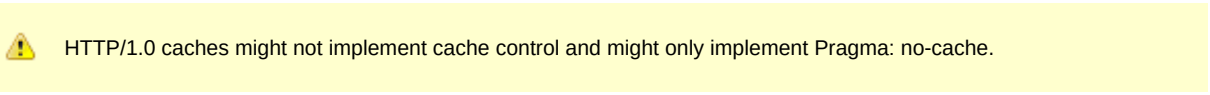
FastArticleCacheTime	You can enable caching of a full article using this option:
	`FastArticleCacheTime=xxx`
	where xxx is the lifetime in minutes for the full page cache files. The default setting is 0, meaning that the feature is disabled.Typical settings with all three Fast cache methods set to two minutes would be:
	`FastFrontPageCacheTime=2`
	`FastSectionCacheTime=2`
	`FastArticleCacheTime=2`
	`CC_GEN=max-age=0, s-maxage=0`
	`EX_GEN=0`
	For more information on Cache-Control headers, please see the section CacheHeaders

FastFrontpageCacheTime	You can enable caching of a full front page using this option: `FastFrontPageCacheTime=xxx` where xxx is the lifetime in minutes for the full page cache files. The default setting is 0, meaning that the feature is disabled.For an example of how this is used, see **FastArticleCacheTime**.
FastSectionCacheTime	You can enable caching of a full section using this option: `FastSectionCacheTime=xxx` where xxx is the lifetime in minutes for the full page cache files. The default setting is 0, meaning that the feature is disabled.For an example of how this is used, see **FastArticleCacheTime**.
Frontpage_Cachetime	Optional. Sets the cache time for the front page. It is set like this: `Frontpage_Cachetime=180`
Gallery_Cachetime	Optional. Sets the cache time for galleries. It is set like this: `Gallery_Cachetime=720`
HostNameInFilename	Determines that the host name with which the site is accessed is used in the cache file name. This function is useful when you use one website for multiple functions that generate cache files with the same names. The option is enabled like this: `HostNameInFilename=1`
ImageCache	Optional. Sets the cache time for images. This option is used when you have set cache=0 to cache the images on the website during development. It is set like this: `ImageCache=1`
InstanceLimit	This option sets the maximum number of instances that the dll should start before the server gives the message, *Server too busy*. The default is 100. The option is set like this: `InstanceLimit=100`

Misc_CacheTime	Optional. Sets the cache time for miscellaneous files. It is set like this: `Misc_CacheTime=720`
Section_Cachetime	Optional. Sets the cache time for section fronts in minutes. It is set like this: `Section_Cachetime=180`
SessionLimit	Optional. Limits the number of concurrent sessions a user can have on the website. The option is set like this: `SessionLimit=4`

Cache Headers

This section allows you to set the cache control settings for the Cache-Control header field. The Cache-Control header field is used to specify directives that MUST be obeyed by all caching mechanisms along the request/response chain. The directives specify behavior intended to prevent caches from adversely interfering with the request or response. These directives typically override the default caching algorithms. Cache directives are unidirectional in that the presence of a directive in a request does not imply that the same directive is to be given in the response.



Cache control options are documented in RFC2616. For more information please see:

http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html

```
[CACHE]
CC_FRONT=
CC_SECTION=
CC_ARTICLE=
CC_GEN=
EX_FRONT=
EX_SECTION=
EX_ARTICLE=
EX_GEN=
```

|| **CC_FRONT** | Cache control settings for the front page.
CC_FRONT=max-age=xx (in seconds)
||

CC_SECTION	Cache control settings for the sectionfront. CC_SECTION=max-age=xx (in seconds)
CC_ARTICLE	Cache control settings for story pages. CC_ARTICLE=max-age=xx (in seconds)
CC_GEN	The fallback setting if the options above do not exist.
EX_FRONT	The expire time for the front page in minutes.
EX_SECTION	The expire time for the sectionfront in minutes.
EX_ARTICLE	The expire time for the story pages in minutes.
EX_GEN	The fallback expire time if the options above do not exist. (in minutes)

The cache control string can be, for example:

```
max-age=300, s-maxage=300.
```

Fast Cache

DontUseFastSectionOnSections	When running fastsectioncache with reader-specific content like newslists based on users' behavior, those newslist tags were cached. Now there is a blacklist you can set in the publicus.ini file that determines which sections should not be handled by fastsectioncache even if it is enabled. It is a comma separated list of the categories you do not wish to fullpage cache. `DontUseFastSectionOnSections=category1,category2`
AllowFastSectionOnSections **From version 6.2.0sr29 and 6.3.0sr2**	It is now possible to only enable static section cache for a specified list of sections. if a section is defined in both DontUseFastSectionOnSections and AllowFastSectionOnSections then the section will not be cached using static section cache. The list is a comma separated list of the categories you do wish to fullpage cache. `AllowFastSectionOnSections=category3,category4`
DisableFastArticleOnCategories	It has bow been made possible to disable static caching of stories on specific categories and templates `DisableFastArticleOnCategories=category1,category2`
FastPageCacheReplaceSessionInfo	Since no tags are processed when using static cache files for front page, section and stories, registering user information using `<pbs:reguserinfo>` does not work. When using static cache files, the reguserinfo tag must be replaced with `<pbs:#sessioninfo#>`. Enter `FastPageCacheReplaceSessionInfo=1` to replace the session info. The default is 0.
IfNotExistUseFromYesterday	Use the cache from yesterday if there is none for today. Start builing a new cache file. the default is 0
NoCacheWhenUpdFCacheFileInBackground	Do auto nocache when updating static cach files. The default is 0
UpdFCacheFileInBackground	Update static pages in the background by calling the current URL + updfCache=1. The default is 0.
Webserver=AWebServerName	The webserver that updates the content in the background. If it is not supplied, the domain setting from the publicus.ini will be used instead.

There are a few things to consider when setting this up:

- This only works with FastSection/Article/Frontpage cache enabled. See earlier.
- The switch NoCacheWhenUpdFCacheFileInBackground can be hard on your database server as it rebuilds all tags on the whole page every time the fastpage cache runs out.
- Webserver is needed in some conditions, and not needed in others, depending on the hosting setup of your site. Omitting this setting in some conditions will result in totally static pages, not updating content at all.
- If your setup is a single webserver directly connected to the Internet you don't have to give it a value.
- If you are behind a load balancer or router of some sort that changes the request host from www.yoursite.com to like loadbalancer1, the webserver-setting must be assigned the name of the domain. For example: www.site.com. If this is forgotten, the site will become static.
- If you are in doubt, you can try to see what the variable <%currentURL%> contains (It's a variable that you can use on an item template or a macro). If it contains www.yoursite.com then everything is fine without the webserver setting. If it contains anything else than your site, set the webserver setting as described above (for example www.yoursite.com).

Do not use http:// since this is added by the code.

CAPTCHA section

CAPTCHA section

The settings control the general <pbs:captcha> tag that renders a captcha which can be used for client-side/scripting validation.

ForceUnpersist	If value is 1, it overrides the &unpersist=N argument on /checkcaptcha, ensuring that users cannot keep submitting using the same captcha when using client-side captcha validation. Default is 0.

CCE section

This section controls settings for the CCE Module. The section has the heading **CCE.**

```
[CCE]
CCEClassifieds=0
ClassifiedsModule=11
Cache=1
ForceCache=1
ForceCacheTime=number of min.
ForceFieldsCache=1
ForceFieldsCacheTime=number of min.
NoMDCache=1
```

Cache	Determines whether caching is enabled. The default is
	`Cache=1`
	To disable caching, enter
	`Cache=0`
	This is not recommended unless you are debugging your pages.

CCEClassifieds	Determines that CCE data will be used instead of classifieds data. This means that CCE will be used when the Classifieds tags (like Pbs classifieds and Pbs classifieddetails) are processed. To use CCE data, enter: `CCEClassifieds=1`
ClassifiedsModule	Determines the module that is used when CCE processes Classifieds tags.

ForceCache	Enter
	`ForceCache=1`
	to force caching of CCE objects. The default is
	`ForceCache=0.`
	It is recommended that the ForceCache setting be used to prevent high CPU usage on the web servers.

ForceCacheTime	Enter a number of minutes between caching intervals. The recommended value for this setting is 3. It is recommended that the ForceCacheTime setting be used in conjunction with ForceCache to prevent high CPU usage on the web servers.

ForceFieldsCache	Enter
	`ForceFieldsCache=1`
	to force caching of CCE fields. The default is
	`ForceFieldsCache=0.`
	It is recommended that the ForceFieldsCache setting be used to prevent high CPU usage on the web servers.

ForceFieldsCacheTime	Enter a number of minutes between caching intervals. The recommended value for this setting is 3. It is recommended that the ForceFieldsCacheTime setting be used in conjunction with ForceFieldsCache to prevent high CPU usage on the web servers.
NoMDCache	Used to disable the metadata cache for CCE. The metadata cache does not store information about custom fields, causing the system to only identify custom fields correctly once every five minutes. This problem can be avoided by disabling the metadata cache.To disable the cache, enter `NoMDCache=1`

CLASSIFIEDSIMAGE section

Under the heading **CLASSIFIEDSIMAGE** you define settings for Classified ad images.

Transparency	Disables transparency in images in the Classifieds system. `Transparency=0`

CLICKSHAREPRODUCTS section

The Product for Clickshare integration must be defined in the product list. Usually, only the default, or product 0, is used. You can also define additional subscription levels, such as a bronze, silver and gold.

```
[CLICKSHAREPRODUCTS]
0=default
2=premium
```

CLICKSHARE section

This section is used to make settings for the Clickshare integration.

The URL and key are provided to you by Clickshare.

```
[CLICKSHARE]
RequestString=https://home.website.com/clickshare/authenticateUserSubscription.do
?[%CSProduct=<%Product%>&%]CSAuthReq=<%CSAREQ%>&CSTargetURL=<%URL%>
CSAREQ=<%TimeStamp%>:<%Session%>[%:<%Hash%>%]
AppendHash=0
Key=12ab34cd
```

AppendHash	If the hash will not be sent in the URL, enter `AppendHash=0`
BypassMinutesOffset	This setting allows an article request to bypass the clickshare authentication redirect based on a given number of minutes after the article's published date. A valid entry must contain an integer that indicates the number of minutes after the article's pubdate. If the value is positive then the article will be locked after the specified number of minutes have past. If the value is negative then the article will be opened after the specified number of minutes have past. The example below bypasses the authentication redirect for two days: `BypassMinutesOffset=2880`

CSAuthReq	This is defined as a variable in the Publicus.ini, consisting of:
	TimeStamp assigned by SAXOTECH Online on the way to iServices server and assigned by iServices on the way back to SAXOTECH Online.
	Session assigned by SAXOTECH Online and stored in cache for the entire length of the login session.
	Hash A key (any combination of letters and numbers (for example: b63e9de2) that we will encode in an MD5 hash and send with our request to the customer's servers. This relies upon a predetermined key, agreed upon by both SAXOTECH and iServices, and defined in the field **Key**. The hash MUST be sent back by the third party to SAXOTECH Online in order for the process to be complete. SAXOTECH Online does not have to send the HASH in the URL, and if that is the case, the Publicus.ini file will have
	`AppendHash=0`
	However, the third party still has to send the HASH back to SAXOTECH Online in the
	`CSAuthResp`

CSProduct	The Product should be defined in the CLICKSHAREPRODUCTS section.
Key	The key provided by Clickshare.
URL	The URL provided by Clickshare.

COLLECTMOSTREADSTATS section

CollectMostReadStats is a component that collects statistics of how many times stories are shown, e-mailed, and printed.

After a specified duration of time, it flushes the information in log files. This is an improved alternative to the LiveArtStat scheme for collecting story related statistics.

It is recommended to use CollectMostReadStats instead of LiveArtStat. To enable CollectMostReadStats, set LiveArtStatUpdate=2 under the **GENERAL** section of Publicus.ini. To configure the settings for CollectMostReadStats specify the following settings in the **COLLECTMOSTREADSTATS** section:

```
[COLLECTMOSTREADSTATS]
LOGDIR=logdir
RUNLOGDIR=runlogdir
FLUSHINTERVAL=N
```

FLUSHINTERVAL	The number of minutes between every flush. The default is 1.
LOGDIR	The directory to which data log files will be written. The default is `/apps/mostread/logs/`

RUNLOGDIR	The directory to which run log files will be written. The default is `/apps/mostread/run/`

CONTENTVALIDATION section

From version 6.2.0

This section determines that the Pbcs.dll prevents SQL or scripting injections in Content fields. The section is used like this:

```
[ContentValidation]
ContentValidation=1
InvalidContentInput=(?-m)(?-i)(?-g).*alert.*\(.*\).*,select,update,drop,script
HaltOnInvalidContentInput=1
ErrorMsg=Your request contains a forbidden substring.
ErrorStatusCode=303
```

And has the following settings:

ContentValidation	Enables ContentValidation.
InvalidContentInput	A list of comma delimited regular expressions that are not allowed in Content fields.
HaltOnInvalidContentInput	Determines whether a request is halted upon invalid input. The default is 0 (do not process the request).
ErrorMsg	The message to be returned to the browser if HaltOnInvalidContentInput is set to 1.
ErrorStatusCode	The code returned to the browser if HaltOnInvalidContentInput is set to 1 (default is 404).

DEBUG section

This section should only be used by SAXOTECH or by self-hosted customers!

DumpNewslist	Exports the database call to a file in the DBLogs folder. Default is 0. `DumpNewslist=1`
PBOWarningTimeout	Sets the number of milliseconds a ReplaceVars parse may last. Default is 7500 milliseconds.
ExtendedPBOWarnings	If enabled (=1) the generated PBOWarnings output the current template code and the variables given to it. Also it outputs the request URL. Default is 0.
ExtendedPBOWarningsCropVariables	If enabled (=1) the generated PBOWarnings Variable value output is cropped and line shifts are removed. Default is 1.

EDITINTERFACE section

These settings control the behavior of the editing interface.

AgendaExport	Controls if the export button should be available in the agenda interface. The option is set like this: `AgendaExport=1`

DefaultPublished	Optional. Determine that new stories are not published by default. To enable this option enter: `DefaultPublished=0`
DefaultRedKat	Optional. Sets the default category for new stories. This option can also be set for individual users in their myPublicus settings. The option is set like this: `DefaultRedKat=NEWS01`
DefaultRotate	Sets the default rotation type in the editing interface. The options are: • 0 = No rotation. • 1 = Rotate within category. • 2 = Rotate for all categories. • 3 = Rotate within group. `DefaultRotate=1`
FrontpageArticles	Optional. Disables the obsolete front page story function. It is set like this: `FrontpageArticles=0`
GraphHeight	Optional. Sets the height for the poll graphs in the editing interface. The option is set like this: `GraphHeight=100`
GraphWidth	Optional. Sets the width for the poll graphs in the editing interface. The option is set like this: `GraphWidth=122`
IndexedILinkSearch	This option can be used to disable the indexed internal link search and use unindexed search instead. The advantage is that you can search stories that have not been indexed yet. The drawback is that you can only search using one word. `IndexedILinkSearch=0`
LastKeyValues	Optional. Controls if last used themes and keywords should be available in the drop-down menus in the story editing interface. The option is set like this: `LastKeyValues=1`
OnlyCounties	Enables the listing of the top level in the Regions field. The option is turned on by this line: `OnlyCounties=0`
PrioDateSpan	Optional. Sets how many days back priorities should be changed when you use the rotate priority function. The option is set like this: `PrioDateSpan=2`

PrioRotateIncrement	Optional. Sets the increment of priorities for the rotate priorities function. The option is set like this: `PrioRotateIncrement=1`
PubTid	Sets the publish time for new stories. The option is set like this: `PubTid=06:00`
PubTid1	Optional. Sets the publishing time for Sundays. PubTid2 sets the publishing time for Mondays, PubTid3 for Tuesdays, and so on. These options are used for overriding publishing for specific week days. See the PUBLISHTIMES section for more information on overriding publishing times.
RedFylke	Optional. Sets which additional regions should be available in the editing interface. The option is set like this: `RedFylke=02,03`
RedTid	Optional. Sets the time of day at which the default date will switch to the next day. The option is set like this: `RedTid=23:59`
RetainDateDefault	Optional. Controls if the Retain date property on the story should be set by default. The option is set like this: `RetainDateDefault=1`
ShowFactstitle	Optional. Controls whether the title from the main fact box should be displayed in the story list with the title for the story. The option is set like this: `ShowFactstitle=1`
TaxonomyInclude	When you are using taxonomy, this option must be set to determine the top levels that will be visible in the editing interface. The option is set like this: `TaxonomyInclude=1,2,3`
ThemeKeywordRestriction	Optional. Allows you to restrict the creation of new themes and keywords to administrators. The option is set like this: `ThemeKeywordRestriction=1`
UsePublishControl	Optional. Enables the Publish control function. The option is set like this: `UsePublishControl=1`
UseHTMLEditor	Optional. Enables WYSIWYG editing of stories. The option is set like this: `UseHTMLEditor=1`

EVENTSDEF section

This section is used for creating a link between a guide ID and an event. This allows you to list events for an entry in the guide database.

```
[EVENTSDEF]
GuideID="GuideID",40,1
```

EXTLOGIN section

Extended login enables integration with a third-party payment system to control login to your web sites.
To enable Extended Login make the following setting in the **EXTLOGIN** section:

```
[EXTLOGIN]
Enabled=1
AllowCrawler=1
AccessControl=1
TargetNamespace=[URL to targetNamespace of WSDL]
WSDL=[URL to 3rd party webservice WSDL]
```

AccessControl	Used to enable Access Control, which is used to limit access to stories or photo galleries. To enable Access Control, enter: `AccessControl=1` To use Access Control, you must also enable the AccessControl setting in the **MODULES** section and the ArticlePayment setting in the **GENERAL** section, as well as the Enabled, WSDL and TargetNamespace settings in this list.

AllowCrawler	Enter
	`AllowCrawler=1`
	to determine that crawlers can index protected stories when using Access Control. The default is
	`AllowCrawler=0.`
	Only crawlers defined in the **CRAWLERS** section of apps_pbcsDb.ini are allowed to index stories. It is possible to control whether search engines are allowed to cache the stories. This can be controlled using one of the following two options:
	1. Add the following to the header part of story page:
	`<meta name="robots" content="noarchive">`
	2. Add the following to Robots.txt:
	`Noarchive: /`

ArticleFreeInterval	Determines how long (in minutes) the story is free and available to all web guests. Enter:
	`ArticleFreeInterval=X`
	where
	`X`
	is the number of minutes for which the story will be free.

Enabled	Enables Extended Login. Must also be used in conjunction with Access Control (see above.
EnableGoldLevel	With the Subscription functionality, there are four different levels of payment status on a story (free, temp.locked, locked and gold locked). It is possible to disable the highest level of protection from the user interface by making the following setting in the **EXTLOGIN** section: `EnableGoldLevel=0.` The default value is 1.
EXTLOGINCOOKIENAME	The name of the additional cookie that is created and maintained by SAXOTECH Online when using extended login. The value of the cookie is the number of seconds since midnight, January 1, 1970. The domain of the cookie is defined using the **DomainCookie** setting in the Publicus.ini file. If the **DomainCookie** setting is not available, the current domain in the URL is used. The default is empty.
EXTLOGINLIFESPAN	The expire time (current time + number of days) of the cookie. The default is 180.
TargetNamespace	Used for Access Control (see above). Enter: `TargetNamespace=URL to targetNamespace of WSDL`
WSDL	Used for Access Control (see above). Enter: `WSDL=URL to 3rd party webservice WSDL`
WebServiceRequestTimeOut	It is possible to set timeouts on requests for the Access Control web service. This can be done by adding the following setting: `WebServiceRequestTimeOut=[TimeoutInMilliSeconds]` The default is 5 seconds)

EXTRAFIELDDEF section

Used to setup extra field group definitions. For example:

```
ExtraFieldDefs=MusicReview, MovieReview
```

For each definition, there must be a separate EXTRAFIELDDEF section in the Publicus.ini file.

Example

```
[ExtraFieldDef_MovieReview]
```

In this section you can make a setting like the following:

```
[ExtraFieldDef_MovieReview]
EditExtraFields=movie_titel,movie_instructor,movie_countryoforigin,movie_genre,movie_manuscript,
movie_starring,movie_sorttext
```

The list of ExtraFieldDefs you have set up in the GENERAL section section will appear in the edit interface (story properties). By selecting one ExtraFieldDef from the list in the edit interface, only those EditExtraFields which are defined in the respective section will appear in the story edit tab.

ExtraFieldDef and customizable edit interface for Light ID

Group field definitions can be used with the customizable edit interface for light ID users:

```
ExtraFieldDef_<extra fields group>
LightEditExtraFields=<comma-list of fields>
```

EXTRAMENUCPEDITORIAL1 section

To display the link to the **NewsletterPro** custom edit screen in the **Edit** menu, add the following to the Publicus.ini file:

```
[EXTRAMENU_CP_EDITORIAL1]
caption=NewsletterPro
url=/apps/pbcs.dll/exec?name=NewsLetterPro_Top
target=topp2
```

FACEBOOK section

These settings are available from version 6.4.0

The Facebook section enables you to define the default settings for Facebook single signon. With Facebook single-signon it is possible to connect a Facebook account to a Saxotech Online user and thereby allowing sigle-signon using Facebook login. A query parameter has been added to reguser action to indicate that a user registration should connect a Facebook account to a Saxotech Online user Advanced login

APIKey	This key is generated by Facebook and a new key is needed for each site (domain name)
Secret	This key is generated by Facebook and a new key is needed for each site (domain name)
ShowFacebookID	Show Facebook ID on users in edit interface. Default is 0

More information about generating the keys and implementing sigle-signon can be obtained by contacting Saxotech A/S and on Facebook.

Example:

```
[Facebook]
APIKey=9ce8dbc1e5c7f38a7d74a0ce1934e4ca
secret=89e7e00b9980782f7b3c2eb537679174
ShowFacebookID=1
```

FASTSEARCH section

The Online Search powered by FAST InStream search technology is set up in the FASTSEARCH section. To enable the FAST InStream search engine, the following settings must be made:

```
[FastSearch]
UseFast=1
Usesqlonemptyfastsearch=1
AdvancedSearch=0
FieldSearch=title:,staticbody:,rating:,names:,company:,taxonomy:,byline:
SortFields=title,
pubdate,priority,rating,articlenumber,category,news
```

AdvancedSearch	In order to enable extended functionality of FAST InStream for SAXOTECH Online, AdvancedSearch=1 must be set. The use of the advanced functionality of FAST InStream for SAXOTECH Online requires a special license.If the above setting is not made or in case both primary and backup FAST InStream servers are not responding, all searches will search using SQL.

NumberSearchResults	In order to customize the number of results returned, the following variable must be used: `NumberSearchResults=50` This variable also determines the number of results in forum and image search. Default is 200.
SortFields	A list of the fields by which you can sort search results. It is possible to sort search results by multiple fields, including combinations like rating **pubdate** or **pubdate** rating. For example: `SortFields=title,pubdate,priority,rating,articlenumber,category,news, "rating pubdate","pubdate rating"`
UseFast	Determines whether the FAST engine should be used when searching.
UseSqlOnEmptyFastSearch	Makes it possible to fallback to MSSQL when the result on the FAST search is empty. The variable **<%SearchProvider%>** will contain either FAST or MSSQL strings to indicate which database performed the search. To enable this, add the following: `Usesqlonemptyfastsearch=1` The default setting is 0.

When using the SAXOTECH Online Search powered by FAST, it is possible to customize Boolean and proximity operators in search expressions. This is done in the SEARCHSTRINGDEFS section.

FORUM section

Settings for forums are defined under the heading FORUM.

CaptchaID	The <%ID%> in the captcha object file.
Captcha	The user's solution to the captcha. Protects forums using a captcha using the <pbs:captcha> tag. To enable this setting, enter `Captcha  = 1` The default is 0. The form submitted must contain the CaptchaID value. In case the user submits an incorrect solution, the dll returns the preview page instead. Therefore, a special variable, **<%ValidCaptcha%>**, has been added to the ForumNewMessage.pbo object file that is only available when the solution was incorrect. The value of <%ValidCaptcha%> is always 0.
CaptchaID	The <%ID%> in the captcha object file.
CaptchaFontColor	The font color for the captcha. Needs to be put in the publicus.ini file in the same folder as the PbcsI.dll. `CaptchaFontColor=RRGGBB`
CaptchaNoiseColor	The noise color for the captcha. Needs to be put in the publicus.ini file in the same folder as the PbcsI.dll. `CaptchaNoiseColor=RRGGBB`
CaptchaBackgroundColor	The background color for the captcha. Needs to be put in the publicus.ini file in the same folder as the PbcsI.dll. `CaptchaBackgroundColor=RRGGBB`
CheckReferer	Checks that the referrer of the request is matched against the domain value in the Publicus.ini file. The default is 0.

ConnectedForumModerated	Determines whether connected forums will be created as moderated forums by default. To enable connected forums to be moderated by default enter:ConnectedForumModerated=1
ForumMailid	Sets the sender name for forum e-mails sent from the web server. The option is set like this:ForumMailID=Gazette Forum
ForumMailSender	Sets the sender e-mail address for forum e-mails sent from the web server. The option is set like this:ForumMailSender=forum@gazette.com
GroupListFilter **From version 6.2.0sr16**	The group drop down will only show the first 1000 groups with latest messages. This also enables a freetext filter to help find groups. Possible values are 0, 1. Default is 0.
MailSubject	Sends an e-mail to posting authors when someone replies to their postings.MailSubject=Your forum message has been replied to
MaxDaysInactive	Controls how long inactive threads should be visible in the overview.MaxDaysInactive=90
ProtectedPost	Enabling this setting determines that unregistered users will be redirected to a login/register page when they try to post a new message. The default is 0, which disables this option. To enable the setting, enter:ProtectedPost = 1
UseLogin	Enabling this settings determines that users must log on before posting messages.UseLogin=1
UserListMaxSize **From version 6.3.0**	Defines the maximum number of users displayed in the forum user list. Default is 200 entries.
VerifyUsers	Enables the Extended Forum. Extended Forum indicates that only users already registered and verified can post new messages. Please see **ForumMessageItem.pbo** and **ForumMessagesMessageThreadItem.pbo**. Enabling this setting requires that **ProtectedPost** is enabled. The default is 0, which disables the option. To enable the setting, enter:VerifyUsers = 1
DoNotSaveLastVisitCookie	Posibility to disable saving the lastvisit cookie. The default is 0. Enter:DoNotSaveLastVisitCookie=1

Forum cache settings are made in the GENERAL section section of Publicus.ini. In order for caching to work on forums, both

```
Cache=1
```

and

```
Forumcache=1
```

must be set in the GENERAL section section.

The Article module and Forum module are integrated. To enable web guests to make comments on article pages (NOT forum pages) without logging in, you must add the following to the FORUM section:

```
ProtectedPost = 1
UseLogin=1
VerifyUser = 1
```

FRONTPAGESUBCLASS section

This section defines the subclasses that should be available for the front page. In the example below, there are two subclasses that are to be used for *breaking news* and *major event.* These subclasses can have different layouts than basic stories. For example, they can have larger titles and photos.

```
[FRONTPAGESUBCLASS]
a=Breaking news
b=Major event
```

GENERAL section

GENERAL section

The GENERAL section contains the system settings for the site. This section is started by the header **GENERAL**.

AdServer	Overrules the host name, which can be an internal IP. To enable this option, enter: `AdServer=http://www.mydomain.com`
AddSecCatToStoryShortUrl	When `AddSecCatToStoryShortUrl=1` the section category is added to the story short URL.The default setting is 1.
Agentcode	Determines which press agency you use. This code is used by the <pbs:agent> tag retrieve the right stories. Valid options are: AP = The Associated Press (North America) http://www.ap.org CP = Canadian Press (Canada) http://www.cp.org NTB = Norsk Telegrambyrå (Norway) http://www.ntb.no RZ = Ritzaus Bureau (Denmark) http://www.ritzau.dk TT = Tidningarnas Telegrambyrå AB (Sweden) http://www.tt.se/start/
AllowChangingPublishedStatusOnNewslistTags	Makes it possible to show stories depending on their status. To enable this option, `AllowChangingPublishedStatusOnNewslistTags=1` The default is 0. This enables the use of the status parameter on the newslist tag. parameter must be set to a set of valid statuses separated by comma. For all statu this option see Status.
AllowForceLevel4OnRWatRollover	This setting ensures that the cache file is updated using the Read Write server, but the cache file has not been created since the last rollover. If set to `AllowForcelevel4OnRWAtRollover=1` the profile_cache table is updated using the Read Write server if the cache file doe exist.
AlwaysCheckMainCatMap	This option will make the category mapping look at the main category before using default object. This function is described in more details in AlwaysCheckMainCatM
AlwaysIncludeGlobalAds	Changes the behavior of the Advertising Module. Ads assigned to a spot in a speci category will no longer override general ads, but share the spot. To enable this opt enter: `AlwaysIncludeGlobalAds=1`
AlwaysUseBylineObject	Determines that the system should always use the byline object file, even if it is no turned on for the category in the edit interface. The option is enabled like this: `AlwaysUseBylineObject=1`
Article_CacheTime	Overrides the default **cachetime** for the article action.
Articlelinefeed	This option determines that an extra ` ` tag will not be output after each paragraph by default. The option is set like this: `Articlelinefeed=0`

ArticleRepositories	Allows you to use the category in which the story displayed and not the category in it is placed when using profiles. This option can only be used when you have a 1 tc relationship between profiles and categories. Categories that are treated as story repositories are specified in a comma separated list: `ArticleRepositories=NEWS,SPORTS`
ArticlePayment	Activates a property on stories with four levels of payment. The option is enable lik `ArticlePayment=1` To use this setting for Access Control, you must also enable AccessControl setting MODULES section section and AccessControl, Enabled, WSDL, and TargetNames settings in the EXTLOGIN section.
AssetExportFilename	Enables asset export from the user interface. To enable this option, enter: `AssetExportFilename=1` The default is 0.
AutoArticleCache	Enables automatic re-caching of articles. If you add `AutoArticleCache=1` to the Publicus ini, the article will automatically be re-cached when the user clicks S and Close in the edit interface.The default is 0.
Avis	Sets the site code, which determines the dataset that is used. Changing this code give you data belonging to another site, or no data at all if the code you enter has r been used. Avis is the Norwegian word for newspaper.
Cache	Turns caching on or off for the whole website. Only sites in development should ru caching turned off. The option is set like this: `Cache=1`
CacheTime	Use this setting to specify a cachetime for the action cache (level 1 cache). By defa value is -1 meaning that the cache is disabled.
CCEExport **From version 6.4.0**	Enables export of CCE objects from edit interface. Default is off.
Channel	Optional. Determines which of the channels in the editing interface the website sho use. The default value is web. Channel can also be set to either PDA or WAP. This can also be set directly in a newslist tag. The option is set like this: `Channel=Wap`
Charset	Optional. Sets the character set output in the header. The default value is ISO-885 This option is only required if your language requires a different character set for displaying all local characters correctly. Eastern European websites require the set ISO-8859-2.
CheckForGuideCatID	Optional. This setting supports using multiple sub-sitecodes as subcategories for G The option is enabled by entering `CheckForGuideCatID=0`
CountryCode	Optional. Controls the country code in the options table.

DefaultDetSkjerSok	Optional. Sets the default region for the Agenda. The option is set like this: `DefaultDetSkjerSok=0101`
DoEventCache	Turns on caching of the framework on an Events page. The option is enabled by e `DoEventCache=1` All Events tags must be run with an exclamation mark when this option is used.
Domain	Optional. Sets the domain for your website. This is used to ensure that all links poi the correct site when you edit on another server or have a cluster of web servers. I like this: `Domain=http://www.domain.com`
DomainCookie	This allows a session to be shared across a domain. In the example, two subdoma mydomain.com, for example *news.mydomain.com* and *www.mydomain.com* can s the same session file. To activate this feature, permanentcookie must be enabled. enable this feature, enter: `DomainCookie=mydomain.com.`

EditExtraFields	Optional. Allows you to determine which extra fields are available in the story editir interface. This option only controls which fields will be visible. Fields that are prese the story can still be shown on the site even if they are not editable.
	• The template is `$templatepath$\objects`
	• The description assumes the setting
	`[GENERAL]`
	`templatepath=Templates`
	and adds the subfolder of "object" which should be "objects"
	• The edit template `article\Edit-ExtraFieldsItem` is the default tem for extra fields, so the users do not NEED to add a new template. Using th setting alone will work just fine.
	The option is set like this:
	`EditExtraFields=quote,slug,page`
	On your site you must create a new folder called **Edit-Extrafields** in /Templates/object. Here you can create new files such as **quote.pbo**, **slu** and **page.pbo**. In these files you can use three variables: **<%DisplayNam <%Name%>** and *<%Value%>*.
	For example:
	`<div><%DisplayName$p%>: <input type="text" name= "<%name%>" size="50" value="<%value%>"></div>`
	`<div><%DisplayName$p%>:`
	`  <select name="<%name%>">`
	`  <option value="F" [%{value=F} selected %]>Female</option>`
	`  <option value="M" [%{value=M} selected %]>Male</option>`
	`  </select>`
	`</div>`
	You will now see these fields in your edit interface.

EditExtraFields

Optional. Allows you to determine which extra fields are available in the story editir interface. This option only controls which fields will be visible. Fields that are prese the story can still be shown on the site even if they are not editable.

- The template is `$templatepath$\objects`
- The description assumes the setting

```
[GENERAL]
templatepath=Templates
```

  and adds the subfolder of "object" which should be "objects"

- The edit template `article\Edit-ExtraFieldsItem` is the default tem for extra fields, so the users do not NEED to add a new template. Using th setting alone will work just fine.

  The option is set like this:

```
EditExtraFields=quote,slug,page
```

  On your site you must create a new folder called **Edit-Extrafields** in /Templates/object. Here you can create new files such as **quote.pbo**, **slu** and **page.pbo**. In these files you can use three variables: **<%DisplayNam <%Name%>** and *<%Value%>*.
  For example:

```
<div><%DisplayName$p%>: <input type="text" name=
"<%name%>" size="50" value="<%value%>"></div>
<div><%DisplayName$p%>:
  <select name="<%name%>">
    <option value="F" [%{value=F} selected
%]>Female</option>
    <option value="M" [%{value=M} selected
%]>Male</option>
  </select>
</div>
```

  You will now see these fields in your edit interface.

EditExtraFieldsPrefix

A prefix for the lookup of edit extrafields templates can now be set. When the prefi set, the system will look for story extra fields templates in the Edit Extrafields folde setting the prefix you can now direct it to the `<Prefix>Edit-Extrafields fol` The prefix can contain backslashes denoting directories. Enter

```
EditExtraFieldsPrefix=<Some prefix>
```

EditRowColors

To determine that stories should have a different background color in the story list according to their status, enter

```
EditRowColors=1
```

The default is 0.

EditShowDates	Determines which of the dates relevant to a story are shown in the story list and the format in which they are shown. You can also determine that some dates should be shown in a hover box. It is possible to show the following dates: • Dato • RetainedDate • LastModified • Created • PublishFrom • PublishToThe format in which the date should be shown is added in paren after each date type. If no format is given, it will default to the short date format.Example: `EditShowDates=Dato(dd.mm.yyyy), LastModified, Created, EditShowDatesHover=RetainedDate (yyy-mm-dd hh:nn), LastModified`
EnableArticleCopy	This adds a copy button to the Saxotech Online editing interface that allows the us copy a specified number of articles. Enter `EnableArticleCopy=1` to enable the functionality. The default is 0.

EnableMultiPriority	Enter
	`EnableMultiPriority=1`
	to enable the functionality. The default is 0.If
	`EnableMultiPriority=1`
	you need to set
	`showref=256`
	to output the **<%\Summary%>** field.

EnableScrollBarForTaxonomy	This enables scroll bars in the Taxonomy box in the story list view of the Edit interfa when the list of taxonomy words is longer than the height of the screen. To enable bars enter: `EnableScrollBarForTaxonomy=1`
EnableSectiontypeSupport	Controls whether the system will check for updates in the category before refreshir cache file. Enter `EnableSectiontypeSupport=1` if the system should check for updates in the category. The default is 1.
EventsClass	Sets the default class for Agenda and Events. The option is set like this: `EventsClass=1`

ExtraFieldDefs	Use this setting to prepare for extra sets of fields to use in the edit interface (story properties). For example: `ExtraFieldDefs=MusicReview, MovieReview` For each definition, there must be a separate EXTRAFIELDDEF section in the Publicus.ini file.

Fontface	Optional. Sets the font for the tag **<pbs:fontface1/2>** and tags that do not use obje files. The options are set like this:
	`Fontface1=Arial,Helvetica,Sans-serif`
	`Fontface2=Arial,Helvetica,Sans-serif`

ForumCache	Turns on caching of the framework on a forum page. The option is enabled by ente `ForumCache=1` All Forum tags must be run with an exclamation mark when this option is used.
Forum_cachetime	Defines the cache lifetime for the forum. If not present, the cache lifetime will be 20 minutes.
Forum_group_cachetime	Defines the cache lifetime when listing groups. This setting overrides the ForumCa setting.
Forum_message_cachetime	Defines the cache lifetime when listing topics and replies. This setting overrides the ForumCache setting.
Forum_theme_cachetime	Defines the cache lifetime when listing themes. This setting overrides the ForumCa setting. ⚠ The settings above define the cache lifetime for the forum as long as there aren't new entries. For example, whenever a new reply is posted, the following cache files will be flush immediately: The reply list containing the new reply The topic list containing the new reply The group list containing the new reply The theme list containing the new reply If a new topic is posted, the following cache files will be flushed: The topic list containing the new topic The group list containing the new topic The theme list containing the new topic
Fylke	Sets the value for region. This setting must match the regions entered into the data The option is set like this: `Fylke=01`
GetContentTimeout	Sets the default timeout period (in milliseconds) for Pbs getcontent. The default val 5000 and it cannot be set to a value below 100 milliseconds. Enter `GetContentTimeout=5000`

GlobalFirst	Optional. Used with the setting GlobalPaths. Enter `GlobalFirst=1` to detrmine that the system will use global templates before using local templates.
GlobalPaths	Optional. Sets the global paths that should be used for templates. This option is on when you have a global template repository for multiple sites. The option is set like `GlobalPaths=e:\www\globals\regional,e:\www\globals\national` You can also enter **local** in the GlobalPaths to use the local templates. The paths f templates are used in the order that they are given in the GlobalPaths setting.
ImageBorderColor	Used to control the border color of images on the website. It is set like this: `ImageBorderColor=#000000`
ImageCopyright	Used to mark images with copyright information. It is set like this: `ImageCopyright=DailyNews.com`
ImageCopyright_XX	Optional. Sets the copyright text for the sitecode XX. The option allows you to have multiple copyright text strings when you run a shared image server. The option is s this: `ImageCopyright_XX=www.somesite.com`
Imageserver	Optional. Used if you have a separate web server that serves all images on your w It is set like this: `ImageServer=http://images.domain.com`
IniPath	Optional. Only used when you edit on a separate machine in a non-clustered environment. The path must end with a slash, like this: `\\someserver\e\www\www.site.com\`
Language	Sets the language for your website. This option controls the language used in the e interface and date formats on your site. Valid languages are: • DEN = Danish • ENG = English • FRA = Canadian French • GER = German • NOR = Norwegian • POL = Polish • SWE = Swedish
LeaveBlankTitleBlank	Set this value to 1 to determine that title field output will be blank if no title exists. Otherwise, the title will be shown as (No title).
LightEditExtraFields	Define extra fields on the edit interface for light users. `LightEditExtraFields=<comma-list of fields>`

LightNoCache	Used to control whether Community Site pages should be cached or not. Setting
	`LightNoCache=0`
	will make Community Site pages cached. Setting
	`LightNoCache=1`
	will make Community Site pages uncached.

LiveArtStatUpdate	To enable the LiveArtStat, enter:
	`LiveArtStatUpdate=1`
	Alternatively, to enable the CollectMostReadStats functionality for collecting story statistics and the OnlineRobot for inserting the data as an improved option to ArtSt the option to 2:
	`LiveArtStatUpdate=2`
	We recommend using COLLECTMOSTREADSTATS section. By default, both possibilities are disabled (0).

MailID	Sets the sender name for e-mails sent from the web server. The option is set like tl `MailID=Gazette Newsletter`
MailSender	Sets the sender e-mail address for e-mails sent from the web server. The option is like this: `MailSender=webserver@gazette.com`
Mailserver	Determines which mail server should be used for sending newsletters and tips. It is like this: `Mailserver=smtp.domain.com`
MapQuest	Optional. Enables MapQuest features in the system. This option cannot be turned unless you have a contract with MapQuest. The option is set like this: `MapQuest=1`
MaxCCESearch	Controls how many concurrent Classified searches can run. The option is set like tl `MaxCCESearch=2` These settings control how many requests will be sent to the database server. The settings will influence the response times of your website and should only be chan SAXOTECH.

MaxClassifiedSearch	Controls how many concurrent Classified searches can run. The option is set like t `MaxClassifiedSearch=2` These settings control how many requests will be sent to the database server. The settings will influence the response times of your website and should only be chan SAXOTECH.
MaxDaysForNewslistDateRange	Sets the maximum range of days for stories shown in a newslist tag. For example: `MaxDaysForNewslistDateRange=2`

MaxEventsClass	This option is used with **eventsclass** so only events belonging to your site appear. outputs all categories from the **web.emner** from class 0 (the default) up to whateve have defined.This should not be used in the EU hosting environments, since each may have a custom class. This could result in errors. For example,
	`MaxEventsClass=3`
	will output _Sport 4 timesÑ _once for each custom classÑ if it exists once in all of t The option is set like this:
	`MaxEventsClass=1`

MaxEventsSearch	Controls how many concurrent Agenda(/Events) searches can run. The option is s this: `MaxEventsSearch=2` These settings control how many requests will be sent to the database server. The settings will influence the response times of your website and should only be chan SAXOTECH.
MaxForumsSearch	Controls how many concurrent Forum searches can run. The option is set like this: `MaxForumsSearch=2` These settings control how many requests will be sent to the database server. The settings will influence the response times of your website and should only be chan SAXOTECH.
MaxFreeTextSearch	Controls how many concurrent free text searches can run. The option is set like thi `MaxFreeTextSearch=3` These settings control how many requests will be sent to the database server. The settings will influence the response times of your website and should only be chan SAXOTECH.
MaxNewsListSearch	Controls how many concurrent Newslist searches can run. The option is set like thi `MaxNewsListSearch=8` These settings control how many requests will be sent to the database server. The settings will influence the response times of your website and should only be chan SAXOTECH.

Mimetype	Sets the output MIME type of the site. For example: • Mimetype=text/vnd.wap.wml • Mimetype=text/html • Mimetype=application/xml • Mimetype=text/xml
MultiSiteUser	The system allows single user registration in a cluster. When a user has registered site in the cluster, he will automatically have access to all sites in the cluster.To ena this option, enter: `MultiSiteUser=1` By default this is disabled (0).
NewsletterMailid	Sets the sender name for newsletters sent from the web server. The option is set li `NewsletterMailID=Gazette Tip`
NewsletterMailSender	Sets the sender e-mail address for newsletters sent from the web server. The optic set like this: `NewsletterMailSender=webserver@gazette.com`

NewsListForceLevel	Activates support for running newslists with profiles against a Read Only database forces a level higher than 3. The option is set like this:
	`NewsListForceLevel = 4`
	This feature requires that there is a background job running on the Read Write serv which updates the profile_cache every minute.This feature cannot handle multi site/profiles, so the newslist tags with multi site and profiles will still be executed on Read Write database.Example:
	`Current date: 20060620; Site code: 'xx'`
	Currently there are 10 stories with the current date and profile 1000. Journalists wr new stories with tomorrow's date (20060621), which will effect profile 1000. Even th the controlcache table will be updated every time a new story is created and the profile_cache table will be updated with site=xx and profile 1000, the new stories w be inserted into profile_cache table since they have tomorrow's date.

NewslistSmartCache	When `NewslistSmartCache=1` forcelevel 3 will be used. Forcelevel 3 = Update profile_cache table now (using the Only server).
NewsListUseNewEmbedded	Only relevant for sites running versions older than SAXOTECH Online 6.0.1.The embedded newslist query has been optimized since this one was the cause of mar warnings and errors in the dblogs - errors being caused by timeouts.Since rewriting construction of these SQL statements is complex, the old code has been left in the is used by default. To enable using the code that constructs the new embedded ne query, use the following setting in: `NewsListUseNewEmbedded=[0|1]` The default value is 0.

OnlyUseEmbeddedNewsList **From version 6.2.0sr17**	This setting forces the system to use the embedded newslist SQL statement. `OnlyUseEmbeddedNewsList=1` The default value is 0.
NyNorsk	Optional. Setting this option will give you day and month names in Nynorsk when y have set language=NOR. The option is set like this: `NyNorsk=1`

OnlyLocalCommunities	Optional. used by the Agenda and Events Modules, it affects the **<pbs:eventoptio communities> *tag. When this setting is enabled, *<pbs:eventoptions communities>** will NOT list all communities/counties (fylker), it will be limited to th specified in the Publicus.ini as
	`Fylke=01,02...`
	The option is set like this:
	`OnlyLocalCommunities=1`

OriginalUrlInInheritedLink **From version 6.3.0sr13**	Optional. Setting this option will use the original URL in the InheritedLink variable, i of the URL modified by URLRewrite. The option is set like this: `OriginalUrlInInheritedLink=1`
OverwriteNewsListCacheWithEmptyResult	If a newlist SQL failed or generated an empty result, the content from the previous file was used. This caused problems when an empty result was actually expected. can determine whether the cache file will be overwritten even through the SQL resu empty: `OverwriteNewsListCacheWithEmptyResult=1` The default is 1. By default the cache file is overwritten even though the SQL result empty, however if the SQL statement fails (raises an exception) the cache file will r overwritten, and the content from the previous cache file will be used.
PermanentCookie	This option is used to enable permanent cookies if your version of SAXOTECH On Permanent cookies are disabled by default. To enable the feature, enter: `PermanentCookie=1` The PermanentCookie setting only controls the creation of the **PBCSPERMUSERI** cookie. This cookie contains a session ID. As a side effect of this setting, the physi session file is reused if available.The DLL will try to locate the physical session file on this session ID. If it exists, the DLL will not create a new session file and use the existing one. If the physical session does not exist, the DLL will create a new sessi This setting prevents the creation of multiple session files from the same browser s the session ID is kept in **PBCSPERMUSERID** and the physical session file is reuse the same user.
PhotoAlbumRootPath	Specifies the root folder for exported photo album files. For example:PhotoAlbumRootPath=\\imageserver\photoalbumexports

PhotoGalleryExtraFieldDefs	Use this setting to prepare for extra sets of fields to use in the edit interface (photo properties). For example: `PhotoGalleryExtraFieldDefs=LocationReview` For each definition, there must be a separate EXTRAFIELDDEF section in the Publicus.ini file.
PrivateGuide	Optional. Controls sharing of guide data. Setting this option to 0, will enable data fr other sites to be shown on your site. This option is set like this: `PrivateGuide=0`
PrivateLinks	Optional. Controls sharing of links data. Setting this option to 0, will enable data fro other sites to be shown on your site. This option is set like this: `PrivateLinks=0`
PrivatePortal	Optional. Controls sharing of portal data. Setting this to 0 will enable data from othe to be shown on your site. This option is set like this: `PrivatePortal=0`

ReadLimitAccess	Used to determine whether users can see stories to which they do not have acces
	`ReadLimitAccess=1`
	the user can only see stories in the story list to which he has access.If
	`ReadLimitAccess=0`
	the user can see all stories in th story list, but can only edit the stories to which he access.The default setting is 0. Note that access is category-based.

Redfylke	Used to sets the value for additional regions. This setting must match the regions e into the database. The option is set like this: `RedFylke=A1,A2,A3,A4,A5,A6,A7,A8,A9`
ReduceWapStoryLength	Determines whether story output to WAP is reduced to 1100 characters.If `ReduceStoryWapLength=1` story output is reduced to 1100 characters.The default setting is 1.
SaveInvalidStoryCacheFile	Determines whether the invalid story key cache file is created when the story is not found.If `SaveInvalidStoryCacheFile=0` the cache file will not be created if the story is not found.The default setting is 1.

SaxoSync	Allows synchronization with SAXOTECH Editorial. The option is enabled like this: `SaxoSync=1` Each time the article is saved it is flagged for export. If SaxoSyncCategories is not specified, all stories will be synchronized (when saved). See also ThinClientEditor for manual export of articles.
SaxoSyncCategories	You can limit the synchronization to specific categories with `SaxoSyncCategories=CAT1,CAT2` Stories in other categories will not be synchronized.
SecretImagePublishKey **From version 6.4.0**	In order to show unpublished images in the edit interface a secret key must be add Publicus.ini. This secret key is used for verifying that the request comes from the e interface and not the front end. This must be used to enable publish status on imag photo galleries.
ShowInactiveBannersInArchive	Determine whether you want inactive banner images shown in the archive list. If `ShowInactiveBannersInArchive=1` Banners used on inactive banner ads will be shown in the archive. Otherwise, only banners that are not used on any banner ads are shown in the archive.
ShowPhotoAlbumSharedStatus **From version 6.4.0**	Enables shared photogalleries. Default is off
ShowPicturePublishStatus **From version 6.4.0**	Enables publish status on photogallery images. Default is off. See also SecretImagePublishKey in general section to enable this.
ShowTeaserImageIconInStoriesList	Determine whether you want an icon in the stories list to show if a story have a tea image. If `ShowTeaserImageIconInStoriesList=1` is enabled, a small T icon will be shown next to the numbers of pictures attached tc story for stories with a teaser image.
SmartIngres	Optional. Shortens the summary to a complete sentence, instead of outputting "..." you reach the character limit. The option is set like this: `SmartIngres=1`
Stedopt	Optional. Controls the behavior of the **Location** field. Setting `Stedopt=2` will make the location field available in the first paragraph.

StripInvalidURLInput	Governs the filtering away the following strings from URI's after they have been de # script embed applet object to eliminate the risk of cross site scripting attacks. If `StripInvalidURLInput=1` the strings will be filtered. The default is 0.

TemplatePath	Optional. Sets the folder for your templates. The default is
	
	`/mal`
	
	By setting
	
	`TemplatePath=templates`
	
	the folder /templateswill be used instead.

ThinClientEditor	Used to enable the Thin Client Editor (TCE) for exporting articles to SAXOTECH E The option is enabled like this: `ThinClientEditor=1` See also SaxoSync for automatic export of articles.
TitleIsEditLink	To determine that the story title in the stories list is an edit link instead of a preview enter `TitleIsEditLink=1` The default is 1.

URLrewrite	Determines that SAXOTECH Online should write all links in the short form underst
	URLrewrite. URLrewrite must be installed on the server before this option can be
	enabled. Please see URL Rewrite for more information. The option is set like this:
	
	`URLrewrite=1`
	
	If URLrewrite=1, then /Article/Date/Category/StoryId
	If URLrewrite=0, then /Article?AID=yyyyStoryID
	The default is
	
	`URLrewrite=0.`
	
	For example, you could have http://www.site.com/tennessee%20titans point to:
	
	`/apps/pbcs/dll/section?Category=SPORTS04`

UseGuidAsSessionId	Controls whether `GUID` is used as the session ID to make the session ID less pred Enter `UseGuidAsSessionId=1` to use `GUID` as the session ID. The default is 0.
UseHierarchicalProfileSort **From version 6.2.0sr17**	Enter `UseHierarchicalProfileSort=1` to enable sorting of profiles by hirarchi. Default is 0
UseIPTC	Enter `UseIPTC=1` to make the IPTC field available in taxonomy editing.
UseLastModified	Enable this to use the last modified timestamp on stories when updating the Newsl enable it enter: `UseLastModified=1` The default is 0 (disabled).
UseNewStoryLockingScheme	Enables the new locking scheme to ensure that multiple users cannot log into the s story at the same time. `UseNewStoryLockingScheme=1` The default value is 0 (disabled) in 6.0.5. In 6.1.0 and up the new locking scheme i enabled by default.
UseOnlyPHP5	SOSE scripts with the extensions .php or .phs did not get executed when PHP 4 w installed on the web server. setting solves the problem. The option is enabled like t `UseOnlyPHP5=1` The default is 0.
UseProfileBanner	Banners can be linked to profiles, and work in the same manner as categories. The option is enabled like this: `UseProfileBanner=1`
UseProfileInNewslistCacheName	Includes the profile ID in the cache name. If `UseProfileInNewslistCacheName=1` the profile ID is used in the cache name. The default is 0.

UseProfileInStoryCacheName	Includes the profile in the story cache name. By default, the profile is included in th cache file name. To disable the option, enter: `UseProfileInStoryCacheName=0` Using this option you can avoid having several stories cached with the same layou
UseProfileSortCode	Enter `UseProfileSortCode=1` to determine that the Profiletree tag will be sorted by the sort code on the profile ins of alphabetically.
UserSettings **From version 6.4.0**	Determines whether to use the user settings GUI or not.
UseRWforRuban	By enabling this, it is possible to use the RW db for all classified database calls. Th enabled by adding this: `UseRWforRuban=1` This is needed due to some US replication issues.
UseSectionCatInStoryCacheName	Includes the section category in the story cache name. By default, the section cate included in the cache file name. Disable the option by entering: `UseProfileInStoryCacheName=0` Using this option, you can avoid having several stories cached with the same layou
UseSimple404	The system will show the value of UseSimple404 setting. The following template w used instead of the value of USeSimple404 if available. Please note that this templ not be parsed thus it is used as a flat HTML file. To enable this option please add: `UseSimple404=Page not found`
UseSmartBylines	Enables the Smart Bylines functionality. The functionality is enabled like this: `UseSmartByline=1` The options works with **ArticleByline.pbo** in which you can determine the style of primary author byline text.

GOOGLEMAPS section

Domain	The domain is used by the front end (pbcsedit.dll: `Domain=<The domain of the front end>`
Key	The key is used by the front end (pbcs.dll): `Key=<Key supplied by Google>`

See more on how to obtain a key for domain from Google here:

http://code.google.com/intl/da/apis/maps/signup.html

HOSTSITECODES section

This section is used for selecting which site code the image dll should use for image links when a site code is not present. This function is only used when you share one image server between multiple sites.

```
[HOSTSITECODES]
imgsrv.heraldtribune.com=SH
imgsrv.theledger.com=LL
```

IMAGE section

Settings for image quality and borders are defined under the heading IMAGE.

Border	To disable photo borders, set
	`border=0`
	To enable, set
	`border=1`
	If no value is set, the setting defaults to disabled.

BorderColor	This option sets the border color for all images. Border color is set using the HTML hexadecimal code for the color. The lowest value that can be given to one light source is 0 (hex #00) and the highest value is 255 (hex #FF).The default border color setting is #000000, which is black.

Cache	This option determines whether or not images will be cached. If
	`cache=0`
	image caching is disabled. If
	`cache=1`
	image caching is enabled.The default cache setting is 1.

Cachetime	This setting determines the maximum age for cache files. The default is the cachetime setting in the GENERAL section of the Publicus.ini.
Copyright	This setting contains any copyright watermark you want on images. By default, this is blank.
DPIx and DPIy	Theses values indicate the size of the images used. You might set the values if you need to process the images further.If DPIx and DPIy are not set, the standard settings of the machine (96 or 72 DPI) are used. Not setting these values is recommended, if they are not needed for special purposes.Notice that DPI is only available for formats that include this meta information, like JPG and TIFF images. Formats without this meta information (like GIF) will always use 96 DPI or 72 DPI on older machines/images.
PreTitle	This setting determines whether the lead-in text, if there is any, should be added.
Quality	This option sets the compression ratio for your story images. Setting this value low will result in smaller images that are displayed faster. Setting it too low will result in fuzzy pictures. The value that is best for your website depends on the quality and size of the pictures being sent to SAXOTECH Online. It is set like this:Quality=50The default quality setting is 60.Quality can be set from 1 to 100 with 100 resulting in the largest files. Using values above 80 will most likely not give you better quality pictures.

RequestTimeout	Defines an HTTP request timeout in the Publicus.ini file on the image server. It is used with the urlget parameter, and is set in the following way:
	`[IMAGE]`
	`RequestTimeout=5`
	(in seconds)

Server	This contains the host name of any external image server used.

LIGHTUSERSTORYLIST

It is possible to hide certain controls in the story list for light users. Each control has its own toggle setting in the publicus.ini file:

ShowCopyButton	Determine whether light users can copy stories. The default is 1.
ShowExportButton	Determine whether light users can export stories. The default is 1.
ShowFrontPageButton	Determine whether light users can recache the front page. The default is 1.
ShowMultiEditButton	Determine whether light users can multi edit stories. The default is 1.
ShowProfileDropDown	Determine whether light users can filter the story list by profile. The default is 1.
ShowRefreshPrioButton	Determine whether light users can refresh a story's priority. The default is 1.
ShowStoryPrio	Determine whether light users can see the priority of stories. The default is 1.

It is also possible to define another set of extra fields that are used by light users using the **LightEditExtraFields** setting in the GENERAL section.

It is possible to set a default selection in the **Forum** setting of the **Properties** tab through the **<LightID>DefaultForum** setting in the StoryEdit section.

MODULES section

Under the heading **MODULES**, you define which modules are available:

AccessControl	This line enables the Access Control Module. `AccessControl=1` To use Access Control, you must also enable ArticlePayment in the GENERAL section and the EXTLOGIN section; and Enabled, WSDL, and TargetNamespace settings in the EXTLOGIN section.
AdvBanner	This line controls whether a customer is running the Banner Ad module or OAMS. If the value for advBanner is 0, the Banner Ad module is enabled. If the value for advBanner is 1, OAMS is enabled. `AdvBanner=1`
Agenda	This line enables the Agenda Module. `Agenda=1`
Agenda Movies	This line enables the Agenda Movies Module. `AgendaMovies=1`

Agenda People	This line enables the Agenda People Module. `AgendaPeople=1`
Assets	This line enables the Assets Module. `Assets=1`
Banner	This line enables the Banner Module. `Banner=1`
CCE	This line enables the CCE Module. `CCE=1`
Classifieds	This line enables the Classifieds Module. `Classifieds=1`
ClickShare	This line enables the integration with ClickShare. `ClickShare=1`
Company	This line enables the Yellow Pages Module. `Company=1`
ConfigSection	This option is used for turning the configuration section on or off in the admin interface. `ConfigSection=0`
Email	This line enables the White Pages Module. `Email=1`
Extlogin	This line enables the Extended Login Module. `Extlogin=1`
Forum	This line enables the Forum Module. `Forum=1`
Guides	This line enables the Guide Module. `Guides=1`

HostingMeasurementPortal	This line enables the Hosting Measurements Portal. `HostingMeasurementPortal=1`
Images	This line enables the Permanent Photo Gallery Module. `Images=1`
Light	This line enables the Community Site Module. `Light=1`
Links	This line enables the Links Module. `Links=1`
Multichannel	This line enables the Export feature in the Edit interface, including an Export button in the Story List. Enabling Multichannel allows you to export stories using FTP, e-mail, and so on. `Multichannel=1`
Multimedia	This line enables the Multimedia Module. `Multimedia=1`
NetGuest	This line enables the NetGuest Module. `NetGuest=1`
Newsletter	This line enables the Newsletter Module. `Newsletter=1`
OAMS_AdTargeting	This line enables the Ad Targeting functionality in OAMS. This feature requires purchase of OAMS Enterprise edition. `OAMS_AdTargeting=1`
OAMS_EnableCampaigns	This line enables the Campaigns functionality in OAMS. This feature requires purchase of OAMS Enterprise edition. `OAMS_EnableCampaigns=1`
OAMS_EnableClassifiedAds	This line enables the Classified Ads functionality in OAMS. This feature requires purchase of OAMS Enterprise edition. `OAMS_EnableClassifiedAds=1`

OAMS_EnableScriptAds	This line enables Script Ad functionality in OAMS. This feature requires purchase of OAMS Enterprise edition. `OAMS_EnableScriptAds=1`
OAMS_InventoryBasedScheduling	This line enables the inventory based scheduling functionality in OAMS. This feature requires purchase of OAMS Enterprise edition. `OAMS_InventoryBasedScheduling=1`
OAMS_MultiSiteBooking	This line enables multi site booking functionality in OAMS. This feature requires purchase of OAMS Enterprise edition. `OAMS_MultiSiteBooking=1`
Q&A	This line enables the Poll/Contest Module. `Q&A=1`
Results	This line enables the Results Module. `Results=1`
Subscription	This line enables the Subscription Module. `Subscription=1`
Tansa	This line enables the Tansa Spellchecker Module. `Tansa=1`
Taxonomy	This line enables the Taxonomy and Profiles Module. `Taxonomy=1`
Teasers	This line enables the Teasers Module. `Teasers=1`
TemplateTheme	SAXOTECH Online offers the possibility to create template themes that you can switch between. Examples are in times of breaking news or natural disasters. In these cases, a theme that switches the web site to a "slim" version can help ensure that the service is available during periods of very intense traffic. Another example is themes that relate to holidays.Read more about this in the Template Theme section of the SAXOTECH Online User Manual. To activate this functionality, you must set: `TemplateTheme=1`
Timetable	Unknown. `Timetable=1`

TvGuide	This option enables the TV guide. `TvGuide=1`

MULTIPRIORITY section

The Multi Priority function enables you to give a story different priorities, depending on the profile in which it is being shown.
You set up this functionality under the heading **MULTIPRIORITY**. You also need to enable the function using the EnableMultiPriority setting in the GENERAL section.

OnlyShowProfilesWithDefaultwords	Enter `OnlyShowProfilesWithDefaultwords=1` to show only those profiles that have default taxonomy words. The default is 0.

MULTITABSTORYEDIT Section

These settings are used in the MULTITABSTORYEDIT section and they influence the multi tab story edit only. For single tab story edit, please refer to SINGLETABSTORYEDIT section and for general story edit settings refer to StoryEdit section.

SectionOrder **From version 6.4.0**	Determines the order in which the different tabs will appear on the multi tab story edit. `Edit` and `Properties` tabs are required and will be appended the SectionOrder section list if they are missing. Duplicates are ignored. The following tabs are supported `SectionOrder=properties,edit,keywords,connections,teaser,preview,extrapictures` In addition, the `keywords` and `connections` tabs has a combined tab, `keyandconnections`, which can be used as well and it overrides adding `keywords` or `connections` to the list. Also, when `teaser` tab is not specified, the `properties` tab will not display the teaser checkbox.
ShowExtraPictures **From version 6.2.0sr27** **Depricated from version 6.4.0**	Determines whether the Extra Pictures tab will be shown (=1) or not (=0). Default value is 1 meaning the tab is shown. From 6.4.0 **SectionOrder** can be used for specifying whether or not to show the extra pictures, so this setting is to be considered depricated from that version.

NETGUEST section

The **NETGUEST** section is used for setting the names of the custom fields in the editing interface. The names are only used in the editing interface. It is set up like this:

```
[NetGuest]
CustomName1=Name goes here
CustomName2=Name goes here
```

NEWSLETTER section

Make newsletter settings in the **NEWSLETTER** section.

Count	Determines how many stories should be sent for each profile when profiles are used for newsletters. Setting count=0 will make the system use the cache rows setting from the profile.

OBJECTRANKING section

From version 6.4.0

The following two settings in the publicus.ini file define the range of ranking which is allowed

MinRanking	The minimum rank allowed. Default is 1.
MaxRanking	The maximum rank allowed. Default is 10.

OMNITURE_FRONTPAGE_MAPPING section

Enables the ability to use the misc action for presenting a front page when using profiles with Omniture.

For example:

```
[OMNITURE_FRONTPAGE_MAPPING]
1002=url=/templates/dailynewsfront.pbs&category=dailynews&profile=1002
```

In the example above, whenever a URL hits the pbcs.dll in conjunction with the misc action, the Omniture JavaScript pagename variable is set to Frontpage and the profile is set to 1002.

OMNITURE section

In order to enable the Omniture statistics, the following has to be present in the Publicus.ini:

```
[OMNITURE]
CookieDomainPeriods=2
SuiteID=saxotechcom
BuildHierarchy=0
```

The following settings are available in the **OMNITURE** section:

BuildHierarchy	Optional. If enabled, a category tree will be inserted into the Java Script, and will be accessible from SiteCatalyst. If disabled, only the current category on a page (if any), will be inserted into the Java Script, and not the category tree. The default is disabled.
	
	Example 1 - The category tree in the JavaScript file (BuildHierarchy=1):
	
	`s.hier1="Sports, Football, Vic Carucci notes Pittsburghs loyalty:20060310:60310001"`
	
	Example 2 - Without the category tree built (BuildHierarchy=0):
	
	`s.hier1="Football, Vic Carucci notes Pittsburghs loyalty:20060310:60310001"`
	
	There is a tiny tradeoff between performance and getting the category tree built. You will get the most useful statistical information when BuildHierarchy is enabled, and it will probably make no difference regarding performance, as the information is cached with the rest of the page.

CookieDomainPeriods	This setting refers to the number of periods in the domain when the domain begins with "www." For example, the domain "www.mysite.com" contains two periods (..), while www.mysite.co.jp contains three periods. Another way to describe the variable is the number of sections in the main domain of the site (two for mysite.com and three for mysite.co.jp).
SuiteID	If no SuiteID is set, Omniture statistics will be disabled.

UseHostName	It is possible to use a full URL for the **s_code.js** file instead of a relative one. The **s_code.js** file is required when using Omniture statistics.When looking at the page source code for a website that has Omniture enabled, one might see something similar to this when using UseHostName=0:
	`<script language="JavaScript" src="/js/s_code.js">`
	When using UseHostName=1:
	`<script language="JavaScript" src="http://www.mysite.com/js/s_code.js">`

UseOmnitureTag	To disable automatic insertion of Omniture JavaScript, which is necessary for Omiture to generate statistics, you can use the Omniture tag **<pbs:omniturecode>**. You must use this tag with the following setting made in the Publicus.ini file in the Omniture section: `UseOmnitureTag=1` This setting disables the automatic insertion of the Omniture JavaScript. Instead it inserts Omniture JavaScript wherever an Omniture tag is found in the templates. The following actions (on other tags): Tip a Friend, AdClick, PollClick and NewsLetterSignup are not affected by this setting, because they do not use templates.Because Omniture SiteCatalyst will only generate statistics if there is Omniture JavaScript on the pages, we recommend not using this feature except in special circumstances.

OVRSUBCLASS section

The story classes for sectionfronts are defined under the heading **OVRSUBCLASS**. You define the labels for the different classes.

```
[OVRSUBCLASS]
a=Big News
b=Mega News
```

POLL section

In the **POLL** section, you enable settings to prevent poll participants from voting more than once.

```
[POLL]
CheckReferer=1
UseAgentInVoterID=1
UseIpInVoterID=1
UseSessionIDInVoterID=1
MinTimeBetweenVotes=5
MaxNumVotesFromIP=25
```

CheckReferer	Ensures that the vote comes from the domain value defined in Publicus.ini.
MaxNumVotesFromIP	Maximum number of votes from a given IP. This is a problem when there are many users behind a firewall.
MinTimeBetweenVotes	The number of minutes that must elapse between participants from the same IP can vote. This is a problem when there are many users behind a firewall.
UseAgentInVoterID	The participants browser will be part of the voter ID.
UseIpInVoterID	The remote IP address will be part of the voter ID.
UseSessionIDInVoterID	The session ID will be part of the voter ID.

PRIOROTATEGROUPS section

Under the heading **PRIOROTATEGROUPS** you define the groups for rotation of story priority. The groups are defined like this:

```
1=NEWS01,NEWS02
2=NEWS03,NEWS04
3=ARTS,ARTS01,ARTS02
```

These settings are used with the other settings for rotation.

PUBLISHTIMES section

The special publishing times section is used for overriding the publishing times for a single day. This is usually done in conjunction with holidays. The section is set up like this:

```
[PUBLISHTIMES]
20040704=13:00
20041025=13:00
20041224=11:00
20041224=15:00
```

On the left side you specify the date in ISO format (YYYYMMDD). On the right side you specify the publish time in 24 hour format using colon as a separator.

QA section

The **QA** section is used for setting options for polls.

AlwaysUseIFrame	Enabling `AlwaysUseIFrame=1` determines that the system will always use an iframe to display the result. Without the option enabled, the result will be shown as a part of the page after the poll has expired.This option also fixes a different issue where you add a poll after the roll over time in the morning. Using the old behavior, you would have to recache the pages containing the poll because SAXOTECH Online did not output an iFrame on the page. With the new behavior, the iFrame will be output but it will be blank (or display the no poll message).
UseIFrameForPollmini	Used to disable the iFrame when rendering mini polls. The default setting is 1. To disable iFrames enter: `UseIFrameForPollmini=0`

Extra fields
You can define three extra fields for the Q&A Module. These fields are added to the list of available information fields when you create a poll or a quiz.

```
[QA]
CustomField1=Horoscope sign
CustomField2=Hobbies
CustomField3=Favorite food
```

SEARCHSTRINGDEFS section

When using the FAST search engine, it is possible to customize Boolean and proximity operators in search expressions. This is done in the **SearchStringDefs** section in the following way:

```
[SearchStringDefs]
ValidChars=¾,¿,?
WildCards=*
AndTokens=og,&
OrTokens=eller,|
NotTokens=ikke,!
NearTokens=n¾r,~
ContainsTokens=indeholder,¤
GroupStarts=[
GroupEnds=]
PhraseDelimiters='
```

Default	The following default values are used:
	`[SearchStringDefs]`
	`ValidChars=¾,¿,?,?,?,?,?,?`
	`WildCards=*`
	`AndTokens=and,+`
	`OrTokens=or,|`
	`NotTokens=not,-`
	`NearTokens=near,%`
	`ContainsTokens=contains,#`
	`GroupStarts=(`
	`GroupEnds=)`
	`PhraseDelimiters="`

Valid chars define which characters should be considered legal alphabetic characters beside the letters 'a' to 'z.' This way it is possible to customize the search string parser to any language. Non-alphabetic letters should be avoided since they can destroy the functionality of the parser.

Wild cards are the characters that are used as wildcards in searches. Since these are dependant on the search engine used, they should not be altered unless it is found highly necessary.

The remaining tokens are operators or delimiters. The general rule for defining these is that they must be a single character operator that can be used with no space such as **Bush+Clinton** or word operators that will be delimited with spaces, such as **Bush** and **Clinton**.

- Single character operators should not be the letters 'a' to 'z' or in the lists ValidChars and WildCards.
- Word operators should only contain the letters 'a' to 'z' or the letters in ValidChars.
  An arbitrary number of operators of both types can be put in the lists as alternatives for the operators/delimiters.

The default tokens correspond to a high degree to the old syntax of search strings. One difference is the existence of the Contains token that is FAST InStream specific. The syntax of using the Contains token is: `<fielddef> <Contains token> <Search string>`, for example **Byline#Gore**. With this syntax, it is possible to search in specific FAST InStream-fields. Since the FAST InStream-fields require some knowledge about the structure of FAST InStream, this field will primarily be useful for people with insight in the FAST InStream data model used in SAXOTECH Online.

In addition to the standard search criteria, the Search string parser is able to handle a section added to the end of a search string. This so-called non-parse section contains code that should not be parsed such as search conditions to add to the search string. This can be useful to parse in a URL when searching in pre-existing results, for example when navigators are used. The search string parser uses encryption of this section.

SECTIONSUBCLASS section

The sectionsubclass section defines the subclasses that should be available on the sectionfront. It is set up the same way as the FRONTPAGESUBCLASS section.

```
[SECTIONSUBCLASS]
a=Breaking news
b=Major event
```

SINGLETABSTORYEDIT section

It is possible to edit stories on a single tab instead of using multiple tabs. This feature is enabled/disabled based on user type and site,

where user type is either normal or light. The following settings are used in the SINGLETABSTORYEDIT section. For multi tab story edit, please refer to MULTITABSTORYEDIT Section and for general story edit settings refer to StoryEdit section.

SectionDefaultExpanded	Determine whether the sections will be expanded or collapsed by default. `<list of 0's and 1's matching the order>` O is the default.
SectionOrder	Determines the order in which the different section will appear on the single tab story edit. `Edit` and `Properties` sections are required and will be appended the SectionOrder section list if they are missing. Duplicates are ignored. The following sections are supported `SectionOrder=properties,edit,keywords,connections,teaser,extrapictures` In addition, the `keywords` and `connections` sections has a combined section, `keyandconnections`, which can be used as well and it overrides adding `keywords` or `connections` to the list.
SingleTabForLightUsers	To determine that light users will use single tab editing enter `SingleTabForLightUsers=1` The default is 0, or multitab editing.
SingleTabForNormalUsers	To determine that normal users will use single tab editing enter `SingleTabForNormalUsers=1` The default is 0, or multitab editing.

Below is an example in which all available sections are added to the single tab (**Edit** and **Connections** sections are initially expanded, the rest are collapsed):

```
[SINGLETABSTORYEDIT]
SectionOrder=edit,properties,connections,keywords,extrapictures,teaser
SectionDefaultExpanded=1,0,1
```

StoryEdit section

These settings are available from version 6.2.0

The **StoryEdit** section enables you to define the default settings for article properties. This saves time for your users, because properties will have the default settings selected automatically.

AllowAddingParagraphs	Determine whether users will be allowed to add additional paragrpahs on the Edit tab.
AllowDeletingParagraphs	Determine whether users will be allowed to delete paragrpahs on the Edit tab.
DefaultArticleBlockCount	Determine how many paragraph fields will appear by default on the Edit tab. Enter `DefaultArticleBlockCount=[x]` The default is 5.
DefaultArticleSource	NA (default), Paper or Electronic
DefaultEmbedMultiMedia	List (default), Left or Right

DefaultExtraFieldsType	The name of the ExtraFields definition section you want to use. Default is none.
DefaultForum	None (default), Story or Theme
DefaultFrontpageClass	Determine the default frontpage class for stories. Enter `DefaultFrontPageClass=<class name>` The default is no default.
DefaultKicker	0 or 1, if the setting is omitted the default is 0
DefaultLocation	0 or 1, if the setting is omitted the default is 0
DefaultNewsletterChannel	0 or 1, If this setting is omitted the default is 1
DefaultPDAChannel	0 or 1, If this setting is omitted the default is 1
DefaultPublishControl	0 or 1, if the setting is omitted the default is 0
DefaultPicPublishControl	0 or 1, if the setting is omitted the default is 1
DefaultPrioRotate	None (default), Category, AllCategories or CategoryGroup
DefaultPublished	0 or 1, if the setting is omitted the default is 1 . This setting overrides the GENERAL DefaultPublished setting.
DefaultRating	0 or 1, if the setting is omitted the default is 0
DefaultSMSChannel	0 or 1, If this setting is omitted the default is 0
DefaultSectionClass	Determine the default section class for stories. Enter `DefaultSectionClass=<class name>` The default is no default.
DefaultStoryClass	Determine the default story class for stories. Enter `DefaultStoryClass=<class name>` The default is no default.
DefaultStoryFlow	The name of the flow you want as standard, if any.
DefaultStoryPriority	Set this to a value you would like as default. Its default is set to nothing.
DefaultStoryType	Standard (default), FrontPage (if [GENEREL]FrontpageArticles=1 is used), Note or SubStory
DefaultTeaser	0 or 1, if the setting is omitted the default is 0
DefaultWAPChannel	0 or 1, If this setting is omitted the default is 1
DefaultWebChannel	0 or 1, If this setting is omitted the default is 1
ExtraFieldsPlacement	The extra fields on the Edit tab can be moved around in the Edit section, using this setting. Enter `ExtraFieldsPlacement=(top|right|bottom)` where • Top is above the story body (default) • Right is below the main image, fact and SMS boxes • Bottom is below the story body.
FactsHTMLEditor **From versions 6.2.0sr30, 6.3.0sr5**	Determines whether the facts edit boxes is to be edited using the HTML editor (=1) or using a text area (=0). Default is to use the text area (=0).

FactsHTMLEditorPtoBR **From versions 6.2.0sr30, 6.3.0sr5**	Determines whether the HTML editor used on fact boxes are to translate <P> tags to tags (=1) or not (=0). Default is to do the translation (=1).
HTMLEditorCtrlAlign **From versions 6.4.0sr1**	Determines where the control button panel is located, 'bottom' or 'top'. 'bottom' is default.
ParagraphHTMLEditor **From versions 6.2.0sr30, 6.3.0sr5**	Same as **UseHTMLEditor**, added for consistency.
ParagraphHTMLEditorPtoBR **From versions 6.2.0sr30, 6.3.0sr5**	Determines whether the HTML editor used on paragraphs are to translate <P> tags to tags (=1) or not (=0). Default is to do the translation (=1).
ParagraphStatsInterval **From versions 6.2.0sr30, 6.3.0sr5**	Paragraph stats can be continually updated. This setting determines the update interval in milliseconds. Default is 5000.
ParagraphTextBoxHeight	Set the default height (in pixels) of the paragraph boxes on the Edit tab. The default is 30.
<LightID>DefaultForumSetting	Determine the default forum setting for light ID users. Enter `<LightID>DefaultForumSetting=N` The default is 0. The values of N can be: • 0 = off • 1 = Story • 2 = theme/category
ShowByline	Determine whether the **Byline** field will be shown on the Edit tab in the edit interface.
ShowLocation	Determine whether the **Location** field will be shown on the Edit tab in the edit interface.
ShowMainFactBox	Determine whether the main fact box functionality will be available on the Edit tab in the edit interface.
ShowMainImageCrop **From version 6.3.0sr11**	Determine whether the main image in the story edit Edit tab can be cropped. Default is off.
ShowNetTitle	Determine whether the **Net title** field will be shown on the Edit tab in the edit interface.
ShowParagraphFact	Determine whether the paragraph fact box functionality will be available on the Edit tab.
ShowParagraphHeading	Determine whether the **Subheading** field will be available for paragraphs on the Edit tab.
ShowParagraphImage	Determine whether the paragraph image functionality will be available on the Edit tab in the edit interface.
ShowParagraphStats **From versions 6.2.0sr30, 6.3.0sr5**	Determines whether the paragraph stats box is to be displayed (=1) or not (=0). Default is not to display the box (=0).
ShowPictureHeading	Determine whether the **Title** field will be available for pictures on the Edit tab.
ShowRating	Determine whether the **Rating** field will be shown on the Edit tab in the edit interface.
ShowRetainDate	Determine whether the **Retain date** checkbox will be shown on the Properties tab in the edit interface.
ShowSMS	Determine whether the **SMS** checkbox will be shown on the Properties tab in the edit interface.
ShowStoryDesign	Determine whether the **Story Design** field will be shown on the Properties tab in the edit interface.
ShowStoryKeywords	Determine whether the **Keywords** section will be shown on the Keywords tab in the edit interface.
ShowStoryMisc	Determine whether the **Miscellaneous** section will be shown on the Properties tab in the edit interface.
ShowStoryPrioRotate	Determine whether the **Prio Rotate** field will be shown on the Properties tab in the edit interface.
ShowStoryPriority	Determine whether the **Priority** field will be shown on the Properties tab in the edit interface.
ShowStoryQwikNavs	Determine whether the **QwikNav** section will be shown on the Keywords tab in the edit interface.

ShowStorySource	Determine whether the **Source** drop-down will be shown on the Propertied tab in the edit interface.
ShowStoryStats **From versions 6.2.0sr30, 6.3.0sr5**	Determines whether the story stats box is to be displayed (=1) or not (=0). Default is not to display the box (=0).
ShowStoryThemes	Determine whether the **Themes** section will be shown on the Keywords tab in the edit interface.
ShowStoryType	Determine whether the **Story type** drop-down will be shown on the Propertied tab in the edit interface.
ShowSuperHeading	Determine whether the **Summary** field will be shown on the Edit tab in the edit interface.
ShowSuperHeading	Determine whether the **Subtitle** field will be shown on the Edit tab in the edit interface.
ShowSuperHeading	Determine whether the **???** field will be shown on the Edit tab in the edit interface.
ShowTitleInTop	When editing stories the story title was only displayed on the Edit tab. Now the title is displayed at the top of the screen and is visible for all tabs. To disable this feature, enter `ShowTitleInTop=0` The default is 1.
StoryStatsInterval **From versions 6.2.0sr30, 6.3.0sr5**	Story stats can be continually updated. This setting determines the update interval in milliseconds. Default is 5000.
UseHTMLEditor	Use this setting to determine whether the HTML editor will be used in the edit interface (=1). Default is off (=0).
UsePublishControl	Use this setting to determine whether publish control will be used.

If any setting is misspelled or omitted, the default is used.

TEMPLATEMAP section

Under the heading **TEMPLATEMAP** template mappings are set up.

Please see Template Mapping for more information.

URLValidation section

This section determines that the Pbcs.dll prevents SQL or scripting injections in the URL. The section is used like this:

```
[URLValidation]
UrlValidation=1
InvalidURLInput=(?-m)(?-i)(?-g).*alert.*\(.*\).*,select,update,drop
HaltOnInvalidURLInput=1
ErrorMsg=Your request contains a forbidden substring.
ErrorStatusCode=303
```

And has the following settings:

UrlValidation	The same as StripInvalidURLInput.
InvalidURLInput	A list of comma delimited regular expressions that are not allowed in the URL.
HaltOnInvalidURLInput	Determines whether a request is halted upon invalid input. The default is 0 (do not process the request).
ErrorMsg	The message to be returned to the browser if HaltOnInvalidURLInput is set to 1.
ErrorStatusCode	The code returned to the browser if HaltOnInvalidURLInput is set to 1 (default is 404).

UserSubmitCCE Section

UserSubmitCCE section

These kind of sections are used to set up CCE object type definitions that are being used along with the UserSubmit section for User Generated Content.

The section is using the format of

```
[UserSubmitCCE_ModuleID_ClassID_TypeID]
```

and it has the following settings available:

CCEModule	Sets the module ID of the definition.
CCEClass	Sets the class ID of the definition.
CCEType	Sets the type ID of the definition.
TaxonomyList	Use to define a list of taxonomy IDs that is available to submitters.

Example of making two type definitions:

```
[UserSubmitCCE_2_2_CarType]
CCEModule=2
CCEClass=2
CCEType=CarType
TaxonomyList=28,144,347

[UserSubmitCCE_3_23_BoatType]
CCEModule=3
CCEClass=23
CCEType=BoatType
TaxonomyList=28,144,321
```

UserSubmit section

Work flow on user generated content

For workflow you can control who appears to have created the content as well as its initial publish status.

Controlling creating UserId

AllowAnonymousSubmit **From version 6.2.0sr17**	If enabled (=1) it is possible for all users to submit content, whereas if disabled (=0) only logged in users can submit content. Default is disabled (0).
AnonymousUserId **From version 6.2.0sr17**	If **AllowAnonymousSubmit** is enabled this setting allows to set who the submitter appears to be by giving this setting a UserId as value. By default anonymous users have UserId 0, which is an invalid UserId making it easy to identify an identity as anonymous in the various templates where variables like CreatedBy are available.
ForceAbonUid **Not available from version 6.2.0sr17**	Overrides the submitted UserID of items submitted by users logged in as subscribers.
ForceUid	Overrides the submitted UserID of items submitted by non-subscriber users. From 6.2.0sr17 only non-subscriber users can submit content.

Controlling publish status

By default, it is possible to use the User Generated Content component to push stories onto the site with any publish status.

To prevent this, the following settings in the **UserSubmit** section are available:

ForceStatus	Same as **StoryForceStatus**
PhotoGalleryForceStatus	Overrides the published status of user submitted photo galleries, 0 = unpublished, 1 = published. Default is 0.
StoryForceStatus	Overrides the published status of user submitted story. 1 = unpublished, 9 = published. Default is 1.

Example:

```
[UserSubmit]
PhotoGalleryForceStatus=0
StoryForceStatus=1
ForceUid=12345
AbonForceUid=Subscriber
```

Which will unpublish stories and photo galleries no matter what the submitters try to do. Also, it will make all subscribers' submitted items to seem to be created by a edit user called Subscriber - and all other items will be given the 12345 user as creator. Anonymous users are not allowed to submit content.

A note, changing the creator will completely disassociate the submitters from their submitted items. They will not be listed as created by the user, and the user will not be able to update or delete the item.

Categories on user generated content

These settings are available from version 6.2.0

For user generated stories and photo galleries it is possible to supply a list of available categories that will be displayed to the user through object files. A setting is available for every specific type and a fallback common setting. Also, it is possible to set a default category that will be used in case no valid category was supplied, again with fallback to a common setting.

PhotoGalleryCategoryList	Use to define a list of category IDs that is available for photo gallery submitters.
StoryCategoryList	Use to define a list of category IDs that is available for story submitters.
CategoryList	Use to define a list of category IDs that is available for story and photo gallery submitters. This setting is used if the type specific setting is not set.

PhotoGalleryDefaultCategory	Use to define a default category for photo galleries.
StoryDefaultCategory	Use to define a default category for stories.
DefaultCategory	Use to define a default category for photo galleries and stories and used if the type specific setting is not set. If this is not set, 'UNKNOWN' or 'UKJENT' is used depending on the language settings.

Example:

```
[UserSubmit]
CategoryList=SPORTS,DOMESTIC,NEWS,CULTURE,LTTE,BLOGS
DefaultCategory=BLOGS
```

Taxonomy words on user generated content

These settings are available from version 6.2.0

For user generated stories, photo galleries, and CCE objects it is possible to supply a list of available taxonomy words that will be displayed to the user through object files. A setting is available for every specific type and a fallback common setting.

PhotoGalleryTaxonomyList	Use to define a list of taxonomy IDs that is available to photo gallery submitters.
StoryTaxonomyList	Use to define a list of taxonomy IDs that is available to story submitters.
CCETaxonomyList	Use to define a list of taxonomy IDs that is available to CCE object submitters.
TaxonomyList	Use to define a list of taxonomy IDs that is available to submitters.

Example:

```
[UserSubmit]
PhotoGalleryTaxonomyList=2,4,23,26,47
StoryTaxonomyList=2,4,23,41,42
CCETaxonomyList=2,4,33
TaxonomyList=2
```

Here, **TaxonomyList** is never used as all types have specific definitions.

Extra fields on user generated content

> **These settings are available from version 6.2.0**

For user generated stories and photo galleries it is possible to supply a list of available extra fields that will be made available to the user through object files. A setting is available for every specific type and a fallback common setting. The setting points to an extra field group definition as described in ExtraFieldDef_section.

PhotoGalleryExtraFieldDefs	Use to allow the extra fields on the given extra field group definition to be attached to the photo gallery.
StoryExtraFieldDefs	Use to allow the extra fields on the given extra field group definition to be attached to the story.
ExtraFieldDefs	Use to allow the extra fields on the given extra field group definition to be attached to the submitted story or photo gallery.

Example

```
PhotoGalleryExtraFieldDefs=LiveGalleryFields
StoryExtraFieldDefs=LiveStoryFields
ExtraFieldDefs=LiveFields
```

Here **ExtraFieldDefs** is never used as both types have specific definitions.

In order to view the same extra fields in the edit interface, use the settings of

```
[Generelt]
PhotoGalleryExtraFieldDefs=LiveGalleryFields
ExtraFieldDefs=LiveStoryFields
```

and if more extra fields are to be available to the edit interface users then it is done by creating another extra field group definition that is a superset of the group definition used for the front-end extra fields.

Comments on submitted stories

> **These settings are available from version 6.2.0**

A setting is available to set if comments are allowed on the submitted stories.

DefaultComments	Determines whether a forum will be associated with submitted stories. `DefaultComments=\[0,1,2\]` (0 no forum, 1 story forum, 2 Theme/Category) The default is 0, meaning moderators have to enable the forum for the individual stories in order to allow comments.

UserVerify section

These settings are used when verifyling users.

DisableLogin **From version 6.2.0sr27**	When a user clicks on a verfication link received upon registering it redirects to the login page if the user is not logged in. DisableLogin enables users to verify their account without being logged in. This is default off.
MailSubject	This value will be used as subject in the mail containing the verification link.

Web server ini file

The web server ini file holds information necessary to connect the website to the web server and the web server to the database server. This file can also be created on a per site basis.

Locations

If this file is put in the c:\winnt or c:\windows folder it needs to be named *publicus.ini* . If its located in the *%driveletter%/inetpub* folder it needs to be called *%sitecode%apps_pbcsdb.ini* or *default_pbcsdb.ini (affects all sites).* The *%sitecode%apps_pbcsdb.ini* will overrule the *default_pbcsdb.ini* file.

Example

> Site MySite.com with site code MY needs different settings than the other sites on the same server. Create a *MYapps_pbcsdb.ini* in the %driveletter%/inetpub folder with the settings you need.

Time zones
There is a setting in this file that controls the time zone information. Please note that the database server and the web server must have the same time.

Example
If the web server is in the Eastern time zone and the website is in the Mountain time zone, you would use this setting:

```
LocalTimeBias=-120
```

Chapter 3 PBCSdb.ini

*_pbcsDB.ini controls the caching of the website and its templates. Settings affecting how the website is cached are placed in this file. Unless you are absolutely sure what the settings do, you should not change these settings. The name of the file may vary. Examples could be Apps_pbcsDB.ini and Default_pbcsDB.ini.

The ini file should not be edited without contacting SAXOTECH.

Placement of the *_pbcsDB.ini file
The ini file is placed in the root folder for the website. Each website's root folder is usually a sub folder of \www and has the same name as the website. This folder is only readable by the customer and the web server.

This chapter contains information about the different sections of the *_pbcsDB.ini file.

- Database connection pool
- Database servers
- Edit template settings
- General cache settings
- GLOBAL section
- Language file caching
- Online Search
- Plugins
- SC_Map
- Template settings
- Thread pool

Database connection pool

The software is using a pool of database connections, and the [CONNECTIONPOOL] section is used for setting it up.

DefaultIdleSeconds	The amount of idle time in seconds allowed before a connection is closed. Default is 300.
DefaultMaxLifeSeconds	The amount of time in seconds a connection is allowed to be closed before it is removed from the pool. Default is 0 (infinite).
EditIdleSeconds	The amount of idle time in seconds allowed before an edit connection is closed. Default is 300.
EditMaxLifeSeconds	The amount of time in seconds an edit connection is allowed to be closed before it is removed from the pool. Default is 0 (infinite).
SearchIdleSeconds	The amount of idle time in seconds allowed before a search connection is closed. Default is 600.
SearchMaxLifeSeconds	The amount of time in seconds a search connection is allowed to be closed before it is removed from the pool. Default is 0 (infinite).

Controlling the size of the pool

From version 6.1.0

In addition there is a way to control

<type>InitialConnections	The initial size of the connection pool. Default is 0.
<type>IncreaseBy	When all connections in the pool is used and another connection is requested this is the amount of connections the pool is increased with. Default is 1.

Here the <type> part of the setting can by one of the following:
DBSERVER, EDITDBSERVER, ADSDBSERVER, STATSDBSERVER, SEARCHDBSERVER, IMPERSONATERWSERVER, and IMPERSONATEROSERVER. An example:

```
[CONNECTIONPOOL]
IMPERSONATEROSERVERInitialConnections=2
DBSERVERInitialConnections=10
DBSERVERIncreaseBy=5
EDITDBSERVERInitialConnections=0
EDITDBSERVERIncreaseBy=5
```

Database servers

Three sections define the database servers:

In the **RODBSERVER** section, the read only database server is set up. The following settings can be made:

IP	Specifies the server name, the IP, or the alias for the read only server.
Trusted	Can be 1 or 0. This parameter can be set to 1 if the web server and the read only database server are in the same domain. If that is the case, there is no need to specify username and password.
UID	The username which is used to connect to the database.
PWD	The password which is used to connect to the database.

In the **SEARCHDBSERVER** section, the search database server is set up. The following settings can be made:

IP	Specifies the server name, the IP, or the alias for the search server.
Trusted	Can be 1 or 0. This parameter can be set to 1 if the web server and the search database server are in the same domain. If that is the case, there is no need to specify username and password.
UID	The username which is used to connect to the database.
PWD	The password which is used to connect to the database.

In the **EDITDBSERVER** section, the edit database server is set up. The following settings can be made:

IP	Specifies the server name, the IP, or the alias for the edit server.
Trusted	Can be 1 or 0. This parameter can be set to 1 if the web server and the edit database server are in the same domain. If that is the case, there is no need to specify username and password.
UID	The username which is used to connect to the database.
PWD	The password which is used to connect to the database.

Edit template settings

The **EditTemplateSettings** section contains the system settings for caching your editing templates.

CachingEnabled	Template caching is enabled by default:
	`CachingEnabled = 1`
	If you want to disable template caching write
	`CachingEnabled = 0`

CacheRefreshInterval	The interval for refreshing a template cache can be modified. By default the setting is 300 seconds (5 minutes): `CacheRefreshInterval = 300` To change this setting, write another number of seconds.
NoOfFileReadAttempts	The number of times the system should try to read the file until it aborts because the file is not there. To change this setting write the number of attempts you need. The default is one attempt: `NoOfFileReadAttempts = 1` Please note that this setting works regardless of whether caching is enabled or not.
FileReadRetryInterval	If NoOfFileReadAttempts is set to a number larger than 1, this setting controls how many milliseconds a request (thread) must wait before retrying file retrieval. The default is 500 milliseconds: `FileReadRetryInterval = 500` To change this setting, write another number of milliseconds.

Best practice: Enable the Edit Template Cache and set the CacheRefreshInterval to 0, since the edit templates are never changed without restarting the dll.

General cache settings

SAXOTECH Online has a fail over system used when caching on a separate server. When the cache server cannot be reached by SAXOTECH Online, a warning is issued. You get a warning in the ExceptionLogs folder when a fail over happens. This file is named **warning_<SessionNumber>*FailedOverToLocalCache*<TimeStamp>.txt**

as shown in this example:

warning_173226033609092_FailedOverToLocalCache_20050512 13-22-20.txt

To set up the fail over system, use the following settings in the **Cache** section:

```
[CACHE]
FailOver_NumberOfConnectionAttempts=1
FailOver_RetryInterval=100
FailOver_ReestablishInterval=0
ForceLocalCache=0
```

FailOver_NumberOfConnectionAttempts	Controls how many attempts should be made before failing over to the local cache. The default is 1 attempt, which is the minimum.
FailOver_RetryInterval	If NumberOfConnectionAttempts is 2 or more, you can specify a slight delay before retrying. The delay is expressed in milliseconds. It is advised to keep this delay very low. The default is 100 ms.
FailOver_ReestablishInterval	If a fail over happens, this parameter controls whether SAXOTECH Online should retry accessing the Cache Server.The `FailOver_ReestablishInterval` is expressed in seconds and the default is 0, which means that no retry will be done. On successful re-access a file with the following name will be written to the ExceptionLogs folder:info_<SessionNumber>*SuccessfullyReestablishedConnection*<TimeStamp>.txt

ForceLocalCache	Can be 1or 0. If set to 1, only the local cache is used. The default is 0. This means that you do not need to specify this setting to use the fail over system.

GLOBAL section

[GLOBAL]

LocalTimeBias

This option sets the time, in minutes, between the web server and the website. The setting is how many minutes the clock on the website is out of sync from the web server.

Example:
If the web server is in the Eastern time zone and the website is in the Mountain time zone, you would use this setting:
LocalTimeBias=-120

The TimeOffset column in the web.dbo.SiteInfo table must also be set to the same offset for the site.

Language file caching

Settings for caching of the language file are made in the **PBOLCache** section.

Enabled	To start caching of the language file, enter: `Enabled=1` This is recommended since PBOL files are not changed without restarting the dll anyway. By default, this functionality is disabled.
NoTranslateMode	Use `NoTranslateMode=0` to disable translation. This is useful for a translator needing to know which strings are used where.

Online Search

The SAXOTECH Online Search powered by FAST search technology is set up in the **FASTSEARCH** section. To enable the SAXOTECH Online Search powered by FAST search engine, the following settings must be made:

```
[FASTSEARCH]
PrimaryHostIP=[IP-address of primary FAST server]
PrimaryHostPort=15100
BackupHostIP=[IP-address of backup FAST server]
BackupHostPort=15100
```

15100 is the standard port used in most setups.

Plugins

Use of more than one `<pbs:newslist>` tag on a page is sometimes required. These tags can be synchronized and you can ensure that stories will not be duplicated on the page. For more information, please see Noduplicates.

In order to use this functionality, an accompanying .dll must be loaded by SAXOTECH Online: **PBCSDupArtFilter.dll**. This .dll is responsible for maintaining the list of stories that will be displayed on the resulting web page. SAXOTECH Online must know the location of this file in order to load the .dll. To facilitate this, the [PLUGINS] section must be added to the *_pbcsDB.ini file:

```
[PLUGINS]
PluginsEnabled=1
PluginsDirectory=Plugins\
```

PluginsEnabled	Set this value to 1 if SAXOTECH Online should load the PBCSDupArtFilter.dll. The default is 0.
PluginsDirectory	Specifies the directory in which the PBCSDupArtFilter.dll is placed.If this setting does not have a value, SAXOTECH Online will look for the .dll in the directory containing the pbcs.dll.

The plugin directory must be a relative path to the apps directory. For example,

```
\PluginsDirectory=Plugins
```

NOT

```
PluginsDirectory=c:\www\inetput\apps\plugins
```

Accompanying the PBCSDupArtFilter.dll should be an .ini file. For **advanced use** it is possible to change some settings in that file. You can optimize control by returning fewer or more stories at any given time. The **PBCSDupArtFilter.ini** can look like this:

```
[EXTRAARTRANGES]
0,5=5
6,10=10
```

When the functionality from the PBCSDupArtFilter.dll is incorporated into SAXOTECH Online, it is also responsible for deciding how many stories will be retrieved in order to achieve the desired number of stories.

The following code provides an example of how to use the SynchID and NoDuplicates parameters:

```
<pbs:newslist days=20 multisite=1 noduplicates=1 site="ap,cj" synchid=1 count=4 objectclass=1>
```

```
<pbs:newslist days=20 multisite=1 noduplicates=1 site="ap,cj" synchid=1 count=2 objectclass=1>
```

In the example above, the first Newslist tag requires 4 stories `(Count=4)` and the second requires 2 `(Count=2).`

After the first tag returns the desired number of stories, in this case 4, the next tag must be processed which requires 2 stories. However, SAXOTECH Online must now make sure that none of these new stories are already present from the first tag. In order to retrieve 2 more stories that are not already present, SAXOTECH Online will need to retrieve more than 2 stories, in case 1 or more is a repeat.

The PBCSDupArtFilter.ini settings above are able to allow access to how many extra stories should be returned. Each setting is based on a particular range of story counts. In the line under the section heading **ExtraArtRanges,** you can specify if a tag requires between 0 and 5 stories (for example Count=3), then instead of retrieving just 5 stories, add an additional 5 stories to the count in case any are repeats. In the second line of the example, if the count needed is between 6 and 10 stories (for example Count=7), then return an additional 10 stories.

SC_Map

Keeping a map file in a cache folder on a central server causes a lot of traffic. Long file names also cause increased traffic. Long file names are used by the `<pbs:newslist>` tag to find the right story to show in the list. However, using the map file can be replaced by Smart Map.

There are two ways for the system to find the right stories to show in the `<pbs:newslist>`, but only Smart Map is recommended for the production environment. The two ways are:

- Caching the map file (can be used for debugging and staging) and using the short names in the cache file. These names are created in a random way.
- Using Smart Map

By default, both features are disabled. If they are both enabled, Smart Map will prevail.

The **SC_Map** section contains the system settings for caching of your site.

WriteMapFileToDisc	By default, WriteMapFileToDisc is enabled. If WriteMapFileToDisc is disabled, Smart Map is automatically enabled. To enable Smart Map, enter: `[SC_Map] WriteMapFileToDisc=0` This will drop writing the Map file to disc and implicitly enable Smart Map.For staging and debugging purposes, writing the Map file to disc can be enabled.
CacheEnabled	When the Map file is written to disc for staging and debugging purposes, reading and writing can be optimized by enabling caching.To enable this option, enter: `CacheEnabled=1` The file must be reread if it meets a long cache file name not already in the cache. This is disabled by default.
SmartMapEnabled	If you have chosen to write the Map file to disc, you can still choose to enable Smart Map. In this case, the way the short filenames are found is changed.To enable this option enter: `SmartMapEnabled=1` This is disabled by default.
AddShortFileNameToStatusLog	When enabled, this adds information about the used (short) cache file name(s) to the StatusLog.These file names can be seen by using the <pbs:!#StatusLog#> tag and by looking at the /status-page. This is disabled by default.To enable it, enter: `AddShortFileNameToStatusLog=1` The cache file names will be listed in the same order as their respective pbs-tags.
AddOnlyIfFileNameSubstituted	If set to 1, items are only output if a file name substitution took place (if the long file name is more than 255 characters and is therefore replaced by a short file name).To enable this filter enter: `AddOnlyIfFileNameSubstituted=1`

Template settings

To reduce disk and network activity, certain logic of reading templates and include files is used. It is enabled through settings in the *_pbcsDB.ini file.

The **TemplateSettings** section contains the system settings for caching the templates of your site.

TemplateThemeRefreshInterval	The interval for refreshing a template theme can be modified. By default the setting is 300 seconds (5 minutes): `TemplateThemeRefreshInterval = 300` To change this setting, write another number of seconds.

CachingEnabled	Template caching is enabled by default:
	`CachingEnabled = 1`
	If you want to disable template caching enter:
	`CachingEnabled = 0`

CacheRefreshInterval	The interval for refreshing a template cache can be modified. By default, the setting is 300 seconds (5 minutes): `CacheRefreshInterval = 300` To change this setting, write another number of seconds.
NoOfFileReadAttempts	The number of times the system should try to read the file until it aborts because the file is not there. To change this setting write the number of attempts you need. The default is one attempt: `NoOfFileReadAttempts = 1` Please note that this setting works regardless of whether caching is enabled or not.
FileReadRetryInterval	If NoOfFileReadAttempts is set to a number larger than 1, this setting controls how many milliseconds a request (thread) must wait before retrying file retrieval. The default is 500 milliseconds: `FileReadRetryInterval = 500` To change this setting, write another number of milliseconds.

Thread pool

To enable or disable the ISAPI thread pool, used the **THREADPOOL** section.

Enabled	Can be 1 or 0. If `Enabled=1` ISAPI thread pools are enabled.
NumberOfThreads	Specifies the number of threads in the timeout time for the search connection.
PbcsIEnabled	Enables thread pools for pbcsi.dll (Image DLL). To enable use the setting as follows: `PbcsIEnabled=1` The default setting is 0.
PbcsINumberOfThreads	Specifies the number of threads for pbcsi.dll (Image DLL). To specify a number use the setting as follows: `PbcsINumberOfThreads=3` The default setting is 30.

Chapter 4 HTML and Includes

Include tags and options are used to include content not only from your site but also from other sites. This section also includes information about the HTML document structure and meta tags.

- Document structure
- Fonts
- Meta tags
  - Article90.pbo
  - ArticleHead90.pbo

- ArticleHead.pbo meta tag
- Pbs filecheck
- Pbs getcontent
- Pbs include
- Server side includes
- Special graphics directory

Document structure

Framework

There are certain tags that should be present in your document. These are shown in the example below.

```
1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
2 <html>
3 <head>
4 <title>SAXOTECH Online Times</title>
5 <meta name="Description" content="Saxotech SAXOTECH Online Demosite">
6 <meta name="Keywords" content="Demosite, SAXOTECH Online">
7 <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
8 <link href="/includes/Style.css" rel="stylesheet" type="text/css" title="Stylesheet">
9 <link href="/includes/TV.css" rel="stylesheet" type="text/css" title="TV Styles">
10 <script language="JavaScript1.2" src="/includes/functions.js" type="text/javascript">
</script>
11 <script language="JavaScript" src="/apps/cookiesearchparams.js" type="text/javascript">
</script>
12 </head>
13 <body bgcolor="#FFFFFF" text="#000000" link="#003366" vlink="#003366" alink="#003366">
14 Your webpage goes here.
15 </body>
16 </html>
```

Tags

1	The document type is set up. This is used to define which HTML tags the browser should expect. If this tag is not present, the document is treated as HTML version 2.0.
2	The HTML tag starts. This tag must enclose the entire document. In the example, the tag is closed on line 16.
3	We start the HTML header. The header contains information about the document and links to resources placed outside the document.
4	Contains the title for the document.
5	Sets the description of the document.
6	Sets keywords for the document.
7	Contains information about the character set used on the page.
8	Links to the main style sheet. By linking to the style sheet, it will be loaded only once. This makes pages load faster.
9	Loads the style sheet for TV/Movies.
10	Contains a link to the shared JavaScript functions used on the website.
11	Contains a link to the cookie management library.
12	Closes the head of the document.
13	Starts the body of the document. Here the background color and the color for links are set.
14	Includes the text that is the document.
15	Closes the body tag.
16	Closes the HTML tag. It should always be the last line in the document.

Fonts

One of the biggest obstacles when trying to keep a uniform layout on a website is to control the use of fonts.

Style sheets

Another way to define fonts is to use style sheets. When you use style sheets, you define how the different items should look in the beginning of the document. An introduction to style sheets can be found at http://www.w3.org/MarkUp/Guide/Style.html. An example of a simple style sheet is below:

```
<style type="text/css">
       A:link,A:visited {
               text-decoration:none;
               color:#566983
       }
       A.example:link,A:visited {
               text-decoration: underline;
               color:#FF0000
       }
       .categoryname {
               font-family: arial,Helvetica;
               font-size: 10px;
               color: #3399cc;
       }
       .title {
               font-family: arial;
               font-size: 16px;
               color: #3E4A5A;
               font-weight : bold;
       }
       #title2{
               font-family: Times;
               font-size: 16px;
               color: #3E8A5A;
               font-weight : bold;
       }
       H1{
               font-family: Times;
               font-size: 16px;
       }
       h2 {
               font-family: Arial, Helvetica;
               font-size: 20px;
               text-decoration:none;
               font-weight : bolder;
               margin : 0;
       }
       P{
       margin-bottom: 0px; margin-top: 0px;
       }
</style>
```

A	Defines the basic look of the anchor tag. The anchor tag is used for links.
A.example	Used when you want to have another look for the link. When you want to use this version, put class="example" inside the A tag.
category name	Used on the category name. Enclose the text with a div tag like `<div class=categoryname>Cateogy</div>`
H1	Used by the tag `<h1>....</h1>` H1 stands for Heading one, and is usually output with a large font size. By retooling the basic HTML codes with styles, older browsers will display the website better. You have the same function for the H2 tag.
P	P stands for paragraph and can be used to enclose paragraphs of text.

title	This style works the same way as the `*categoryname*` style, but it is used for the title.
#title2	This is an example of an ID style. ID can be used instead of class. If you want to know more about the differences between class and ID, please see www.W3C.org.

Publicus.ini
You can define the fonts you want to use in the Publicus.ini. However, this method of managing fonts was created before style sheets were available and should no longer be used. You define the font faces the following way:

```
Fontface1=Helvetica,Arial,Sans-Serif
Fontface2=Palatino,Times,Serif
```

You can define two font faces to be used throughout the site. When you define fonts, you should always remember to define multiple fonts. You should also include the generic font name for users who do not have the font you specified installed on their system.

Here are the generic names and the fonts to which they are usually translated:

```
Sans-serif = Helvetica,Arial,Geneva
Serif = Times, Garamond, Palatino
Monospace = Courier, Terminal
Cursive = Brush Script, Comic
```

In the templates
When you would like to use the font defined in the ini file, you use the following font tag:

```
<font face="<pbs:fontface1>">Text text text</font>
```

SAXOTECH Online will replace the pbs tag with the font information from the ini file.

In the object files
You can also use the defined fonts in object files. You specify the font tags the following way:

```
<font face="<%fontface1%>">Text tex text</font>
```

SAXOTECH Online will replace the tag with the font specified in the ini file.

Font size
The traditional way of controlling font size is by using

```
<font size=#>
```

where # is a number between 1 and 6. In the SAXOTECH Online tags that do not support object files, or where you do not use object files, you can usually set font sizes in the tag.

Colors
In HTML colors are set by:

```
COLOR = #rrggbb
```

The color attribute sets the text color.

```
#rrggbb
```

is a hexadecimal (base 16) red-green-blue triplet used to specify the RGB color.

To set the background color of a web page, use the following

```
<body>
```

tag:

```
<body bgcolor="#rrggbb"> Rest of document goes here </body>
```

Meta tags

Meta tags are set in the HTML Head tag. One of the advantages of setting the HTML Head is that the story will be indexed by search engines.

You need to use the **<pbs:article>** tag twice, once to extract information for the HTML Head and once for the HTML Body.

The pbs / pbo file description you find below does not show all the variables available in each file. These files are only for illustration.

gen_art.pbs

```
<!--the following include will take care of the "HTML head" part of the document.-->
<!--#include virtual="/includes/art_head.inc"-->
<!--the actual article will be shown using the following pbs tag-->
*<pbs:article>*
```

art_head.inc

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<!--this tag is actually the same the one in gen_art.pbs but will use a different Objectclass
thus a different set of pbo files-->
*<pbs:article objectclass=90>*
</head>
```


- Article90.pbo
- ArticleHead90.pbo
- ArticleHead.pbo meta tag

Article90.pbo

```
<!--You must remove all HTML tags in the Head and Summary values-->
<title><%Head%></title>
<meta name="Description" content="<%Summary$t%>">
```

ArticleHead90.pbo

```
<!--the Head variable in Article90.pbo uses this file-->
<%Heading$t%>
```

ArticleHead.pbo meta tag

```
<!--it seems like some crawlers are looking for <h1> tags in the body of a document-->
[%<h1><%Heading%></h1>%]
[%<%Heading2%>%]
```

Pbs filecheck

<pbs:filecheck>

You can also include images based on categories. In this way, if an image does not exist, you have a fallback image. If you use this tag too many places, your site will be very slow.

Example

```
<img src="<pbs:filecheck filename="/images/logo_#category#.gif" fallback="
/images/logo_default.gif">">
```

Options

#Category#	This variable will be replaced with the category ID of the category you are viewing.
Fallback	This field specifies the fallback image if there is no image for the category you are viewing.
Filename	The name of the file that should be included.

#MainCategory#	This variable will be replaced with the category ID of the main category for the category you are viewing.
	Placing
	`*pboscript*`
	in the tag enables you to use pboscript functions in the filename. Example:
	`<pbs:filecheck pboscript filename="/graphics/logo[%{ActiveZone}-french%].jpg">`

#SectionCate#	This variable will be replaced with the category ID of the sectionfrontÕs category and not the category for the story.
Pboscript	

Offsite image include

This function allows you to include a file from another website and use image processing functions on the image. This function is often used when including images from web cameras.

The tag looks like this:

```
<img src="/apps/pbcsi.dll/urlget?url=http://webcam/big.jpg&MaxW=120&Q=100&noborder" border="0">
```

Options

Image options	The function supports all the image options. For more information, please see Image settings.
URL	Sets the URL of the image you would like to download.

Other uses

You can also use this function to process local images. The tag can look like this:

```
<img src="/apps/pbcsi.dll/urlget?url=/graphics/cl_<pbs:categoryid>.jpg&MaxW=120&Q=100&noborder"
border="0" alt="" align="right">
```

Pbs getcontent

<pbs:getcontent>

This function allows you to include a file from another website. If the operation fails, the system will leave a comment about how long it took and the URL that could not be downloaded.

Example

```
<pbs:!getcontent interval=15 cacheid="CLICKTV" url="http://www.clicktv.com/index.asp?cid=AM1">
```

Options

Cacheid	This is the local file name, which should be used for caching. The default setting is the **URL** option. This option is optional.
Cachetime	Controls how long the pbs:getcontent tag is cached in seconds. The default setting is -1, indicating that it does not use any cache. This option is optional.

Imgsrc	Adds a string before x in
	`src="x"`
	of the URL response content. This is used to change local, relative paths in the HTML of the external site to point to the external site. Default is blank. This option is optional.An example is to change
	`src="/apps/g/image.jpg"`
	to
	`scr=http://www.dailynews.com/apps/g/image.jpg by using imgsrc= "http://www.dailynews.com."`

Interval	This option is the same as **Cachetime**. The default is -1 (do not cache). This option is optional.
StatusOutput	If set to 1, the status comment normally added on errors is added on success as well. The default setting is 0. This option is optional.
Timeout	Timeout of the HTTP request in seconds. The default setting is 5. This is optional.
URL	This is the URL that should be pulled from the other site. This option is mandatory.
ConnectTimeout	Limit the time it is allowed for the client to wait for the server to return the 1st byte of the response. Mainly made in order to avoid foreign web servers to take forever before providing answer. Default is 5,000 ms. Available from 6.3.0sr11 and up.

Pbs include

<pbs:include>

A more advanced version allows you to put a category-specific include in a general file. This include type is slower than the standard include, and is performed later in the template assembly. This tag should only be used when there is no other way to achieve the result.

Options

Artupdate	Used instead of cache time. With this option, the file will be updated when changes are made to stories in your database.
CacheQualifiers	Please see Cachequalifiers for more information about caching.
Cachetime	Sets the time the include should be cached before it is refreshed. This option is only used when the tag is running uncached.
Categorymatch	The file will only be included if you are viewing the category specified in categorymatch.
Interval	A synonym for cache time.
Nocache	Disables caching for the file. Usually, the system generates a cache file that is used by all similar includes. When you enable this option, the system caches the content in the main cache file. This option is used like this: `nocache=1`

Useaid	This function includes the date, category and story number in the cache file name. The function can be useful for caching story forum messages (on the story page). The function is used like this: `Useaid=1`
Useprofile	This option is only necessary if you are using profiles and the template changes based on profile. With this option in the tag, a cache file will be created for each profile.
Virtual	Used for specifying the file you would like to include.

Server side includes

Server side includes is a function that allows you to split up your HTML code into multiple files. By doing this, you will only have to maintain code that is used multiple places in one file. This ensures that all pages that should look the same way do look the same way.

Standard includes

SAXOTECH Online supports the standard way of including files. A typical file that includes the standard header and footer would look something like this:

```
<!--#include virtual="/includefiles/header.pbs"-->
Some text that is unique.
<!--#include virtual="/includefiles/footer.pbs"-->
```

The included files contain the information that is present on all pages.

Including files based on category

In addition to including files that are to be used on all categories, it is also possible to have different files included based on the category being viewed. If you use the code:

```
<!--#include virtual="/includes/poll_#category#.inc"-->
```

`#Category#` is replaced by the category ID of the category being viewed. This is the same name that is used for menu links and so on.

If no specific file exists for the category, the system will look for a file with the category name replaced by default. If this file does not exist, SAXOTECH Online will not display anything.

If you are on a subsection, you can use the following piece of code to include a file based on the category ID of the main category:

```
<!--#include virtual="/includes/poll_#maincategory#.inc"-->
```

If no file exists for the main category, the system will fall back to using a file named

```
*default*
```

When template mappings are used

When you map one category to another, the system will first look for the original category, and then the category to which it is mapped. For more information, please see Template mapping#opinion_arts.

For example:

You have created a **gen_art.pbs**, which will be used on all stories. For the categories *Opinion* and *Arts* you would like to include a file in the template. You put the code:

```
<!--#include virtual="inc_#category#.incÓ-->
```

in your file and create **inc_arts.inc**. Instead of creating another copy of the file, which is to be used when the category is *Opinion,* you set the map **opinion=arts** in Publicus.ini instead.

When the system is displaying a story in the *Opinion* category, the system will first look for inc_opinion.inc. If this file does not exist, it will look for inc_arts.inc. If you use this function, you avoid duplicating your code in multiple files.

When profiles are used

In addition to including files that are to be used on all profiles, it is also possible to have different files included based on the profile being viewed. If you use the code:

```
<!--#include virtual="/includes/poll_#profile#.inc"-->
#Profile#
```

is replaced by the profile ID of the profile being viewed. This is the same name that is used for menu links and so on.

If no specific file exists for the profile, the system will look for a file with the profile name replaced by "default." If this file does not exist, SAXOTECH Online will not display anything.

When sectioncat is used

When you use profiles, stories can be placed in another category than the one through which they are displayed. To include files based on the category in which the stories are displayed, use the variable

```
#sectioncat#
```

It is used like this:

```
<!-- #include virtual="/includes/#sectioncat#_header.inc"-->
```

Including files based on zone

When you use zones, the system will automatically try to include a file for the zone you are in.

The regular include:

```
<!--#include virtual="/includes/zone.inc"-->
```

will look for the file /includes/zone.inc when zones are not used. When zones are used, the system will first try to include the file **/includes/$zoneid/zone.inc**.

```
*$zoneid*
```

is the ID you have given to the zone. This ID is usually a three letter abbreviation of the zone name. You must create a folder under the includes folder with the same name as the zone ID. If the folder does not exist, the system will fall back to regular behavior.

Including files based on week day

This option allows you to include different files for different days. The option is used like this:

```
<!--#include virtual="/includes/header_#dow#.inc"-->
*#dow#*
```

is a number between 1 and 7. Sunday is 1, Monday is 2, etc. The default will be used if no file matching the day is found.

Special graphics directory

The special graphics directory is placed outside the base folder for the website. The default location for this folder is **\inetpub\g**. This folder is mounted as a virtual non-browsable, readable folder in the website configuration as **/g**. The folder contains graphic files that are used by SAXOTECH Online and is not editable by the website operator.

The most important files are:

b.gif	Black
t.gif	Transparent
w.gif	White

These files are 1 by 1 pixel and are used for drawing lines and spacing.

In addition, this folder contains other graphics files that are used by the system. You can use these files in your design. These files will always be present on the server.

Chapter 5 Templates

The use of templates is what makes SAXOTECH Online a dynamic publishing solution. The website design is based on the templates. The templates contain codes that pull information from the database, allowing the user to display and control information using tags rather than programming.

Template placement
By default, all templates are placed in the folder **/mal** on the web server. Enabling the option TemplatePath in the Publicus.ini, allows you to use the folder **/templates** instead.

- Export templates
  - article_xml_asset.pbo
  - article_xml_connection.pbo
  - article_xml_paragraph.pbo
  - article_xml.pbo
  - templates_Sms_xml
- Folder structure
- Palm templates
- Setting MIME Types
- Special templates
- Template mapping
- Template names and URLs
- WAP templates
- Zones

Export templates

Export templates can be created to customize the XML output of certain items. Once customized, they must be placed in \objects\export to be used by the system. Customization of these templates will not be overwritten during system updates.

The following export templates are available:

- article_xml_asset.pbo
- article_xml_connection.pbo
- article_xml_paragraph.pbo
- article_xml.pbo
- templates_Sms_xml

Sample code from each template is included in the relevant sections. Unless otherwise noted, only the fields in the template are available.

article_xml_asset.pbo

The following is an example of the article_xml_asset.pbo:

```
<asset>
  <Filename><%Filename%></Filename>
</asset>
```

article_xml_connection.pbo

The following is an example of the article_xml_connection.pbo:

```
<articlekey>
<date><%Date%></date>
<category><%Category%></category>
<articlenumber><%ArtNo%></articlenumber>
<reference><%Reference%></reference>
</articlekey>
```

article_xml_paragraph.pbo

The following is an example of the article_xml_paragraph.pbo:

```
<paragraph id="<%Block%>" html="<%IsHTML%>">
[%
<heading><%Heading%></heading>
%]
<body><%Body%></body>
[%{FactsBody}
<facts align="<%FactsAlign%>">
[%
<heading><%FactsHeading%></heading>
%]
<body><%FactsBody%></body>
</facts>
%][%
<image align="<%ImgAlign%>" mimetype="<%ImgMimeType%>" width="<%ImgWidth%>" height=
"<%ImgHeight%>">
<filename><%ImgFilename%></filename>
%][%
<caption-leadin><%Caption-Leadin%></caption-leadin>
%][%
<caption><%Caption%></caption>
%][%
<photo><%ImgPhoto%></photo>
%][%{ImgFilename}
</image>
%]
</paragraph>
```

article_xml.pbo

In the article_xml.pbo it is possible to use custom fields as variables in the format by using the following:

```
<%FIELDS_FieldName%>
```

The following is an example of the article_xml.pbo:

```
<?xml version="1.0" encoding="[%<%CharSet%>%|%iso-8859-1%]"?>
<!DOCTYPE article SYSTEM "http://www.publicus.com/xml/pbcs5_article.dtd">
<article sitecode="<%Site%>">
<properties priority="<%Priority%>" subarticle="<%Subarticle%>" status="<%Status%>" note=
"<%Note%>">
<date><%Date%></date>
[%{PublishFrom|PublishTo}
<publishtime from="<%PublishFrom%>" to="<%PublishTo%>"/>
%]
<category><%Category%></category>
<articlenumber><%ArtNo%></articlenumber>
<reference><%Reference%></reference>
[%
<subclasstype frontpage="<%FPSubClass%>" section="<%SecSubClass%>" article="<%ArtSubClass%>"/>
```

```
%][%
<forum><%Forum%></forum>
%]
<rating><%Rating%></rating>
<theme><%Theme%></theme>
<keywords><%Keywords%></keywords>
[%
<taxonomy><%Taxonomy%></taxonomy>
%]
<created time="<%Created%>" by="<%CreatedBy%>"/>
<modified time="<%Modified%>" by="<%ModifiedBy%>"/>
[%
<page><%FIELDS_PAGE%></page>
%][%
<section><%FIELDS_SECTION%></section>
%][%
<source><%Source%></source>
%]
</properties>
[%{channels}
<channels WEB="<%ChWEB%>" WAP="<%ChWAP%>" PDA="<%ChPDA%>" TXTTV="<%ChTXTTV%>" SMS="<%ChSMS%>"/>
%]
<url><%URL%></url>
[%
<connections>
<%Connections%>
</connections>
%]
<head>
[%
<superheading><%SuperHeading%></superheading>
%]
<heading><%Heading%></heading>
[%
<heading2><%Heading2%></heading2>
%][%
<netheading><%NetHeading%></netheading>
%][%
<summary><%Summary%></summary>
%][%
<sectionsummary><%SectionSummary%></sectionsummary>
%][%
<byline><%Byline%></byline>
%][%
<location><%Location%></location>
%][%
<image mimetype="<%ImgMimeType%>" width="<%ImgWidth%>" height="<%ImgHeight%>">
<filename><%ImgFilename%></filename>
%][%
<caption-leadin><%Caption-Leadin%></caption-leadin>
%][%
<caption><%Caption%></caption>
%][%
<photo><%ImgPhoto%></photo>
%][%{ImgFilename}
</image>
%]
</head>
[%
<facts align="<%FactsAlign%>">
<heading><%FactsHeading%></heading>
<body><%FactsBody%></body>
</facts>
%][%<assets><%assets%></assets>
```

```
%][%<%Paragraphs%>
%]</article>
```

templates_Sms_xml

sms_xml.pbo

The following is an example of the sms_xml.pbo:

```
<?xml version="1.0" encoding="[%<%CharSet%>%|%iso-8859-1%]"?>
```

```
<!DOCTYPE smsarticle SYSTEM "http://www.publicus.com/xml/pbcs5_smsarticle.dtd">
```

```
<smsarticle sitecode="<%Site%>">
```

```
<properties priority="<%Priority%>" status="<%Status%>">
```

```
<date><%Date%></date>
```

```
<category><%Category%></category>
```

```
<articlenumber><%ArtNo%></articlenumber>
```

```
<created time="<%Created%>" by="<%CreatedBy%>"/>
```

```
<modified time="<%Modified%>" by="<%ModifiedBy%>"/>
```

```
</properties>
```

```
<url><%URL%></url>
```

```
<head>
```

```
<heading><%Heading%></heading>
```

```
[%
```

```
<byline><%Byline%></byline>
```

```
%][%
```

```
<location><%Location%></location>
```

```
%]
```

```
</head>
```

```
<body>
```

```
<content><%Body%></content>
```

```
</body>
```

```
</smsarticle>
```

Folder structure

Your website should have the following folder structure:

/	The root folder for the website. A file called default.htm must be placed in this folder. It must redirect to the front page URL.
/ads	This folder is used for storing ads graphics.
/adlogs	This folder is used for logging by the ad system.
/.cache	This folder is used by the SAXOTECH Online cache engine.
/includes	Files that are included by templates go here.
/light	This folder is used for Community Site sites.
/misc/misc	Files that should be parsed by SAXOTECH Online, but are not templates should be placed in this folder.
/templates	The main folder for template files.
/templates/macros	The main folder for macros.
/templates/macros/<category>	Macros for one category.
/templates/objects	The main folder for object files.
/templates/objects/<category>	Object files that are available for a single category.

Palm templates

To build a functional site for users with handheld devices, you build lightweight templates that are displayable by the browsers built in to the devices. Handhelds can be Palm or WinCe users with the Avantgo browser, or a direct online solution like the PalmVII and OmniSky modems.

Setting up a Palm site

To build a handheld site, you set up an additional site that only contains lightweight templates. This can either be a separate site, or a subfolder on your main site. Please see the examples in the `/palm` folder.

Front page template

The front page is usually handled by the file default.htm, from which you link to the other sections.

Sectionfront

The gen_palmovr.pbs template is used for building the sectionfronts.

Story

The template gen_palmart.pbs template is used for showing the stories.

Setting MIME Types

MIME types are used for identifying content to the browser or application you use for downloading information. By default, these are predefined in the web server based on the file extension. When you download a JPG file, the server automatically sends the MIME type image/jpeg.

You can override MIME type directly in the URL. This is useful if you would like to have your web page treated as XML instead of HTML. As long as you are dealing with text, you only have to specify the last part of the MIME type. Add

```
&mime=xml
```

to set the document's MIME type to text/XML. This allows you to set a different MIME type for special templates. For example, pages that are going to be read by an application, like for an RSS feed.

If you would like to set the MIME type to something different, you can also specify the first part of the string like this:

```
&mime=image%2fgif
```

or

```
&mime=x-application%2fspacemonkey
```

"%2f" is the code for "/." The "/" must be written as

```
%2f
```

Special templates

The following special templates also exist.

Error template

This template is shown when a requested object cannot be found on the server. The file is called **404.pbs** and must be placed in the template folder. In order for this to work, the web server must be set up to pass error requests for the error type 404 to SAXOTECH Online. This must also be done if you are using the Alias function.

This page should not contain any SAXOTECH Online tags. The page is shown for every missing object you have, and it can generate a lot of extra traffic if you have tags on this page.

Refresh page

If you have made changes to your error document and want to refresh them without waiting for the cache to expire, you can use this URL:

```
http://www.yoursite.com/apps/pbcs.dll/error?nocache=1
```

Robots.txt

This file is not a SAXOTECH Online file. It is a standard web server file. The file controls which parts of your website robots will be allowed to access. This file should be present on your website. It must be placed in the root folder of your website.

Template mapping

In the Publicus.ini, there is a **TEMPLATEMAP** section that contains information about which templates are to be used. Use this section to define mappings so that you do not have to make multiple copies of the template files. For example:

```
[TEMPLATEMAP]
_FART=_ART
_FUART=_ART
_UART=_ART
_OVRN=_OVR
opinion=arts
ABOUT_FART=ADVERTISE_ART
ABOUT_OVR=ADVERTISE_OVR
ADMAIL_ART=STATIC_ART
```

All of these options must be placed under the **TEMPLATEMAP** section of the ini file. The different types of templates are described below.

Mapping types

about_fart=advertise_art	With this option you map a specific story template to another category.
about_ovr=advertise_ovr	With this option you map a specific sectionfront to another category.
_fart=_art	This option maps all front page stories to the normal story template. A front page story is a story that has been linked to from a story of the type front page.
_fuart=_art	This option maps all front page subarticle stories to the normal story template. A front page subarticle is a story with the *Subarticle *checkbox marked in story properties, and which is linked to from a story of the type front page.
opinion=arts	This option maps all templates from the *Opinion* category to the *Arts* category, for all template types. When you use include files with the category as part of the name, the system will look for both categories. For more information, please see Server side includes .
_ovrn=_ovr	This option maps all section fronts with notices to the standard overview template. A notice is a story with the notice checkbox marked in the story properties. This template type is obsolete.
_uart=_art	This option maps all subarticles to the story template. Subarticles are stories with the **Subarticle** checkbox marked in story properties. Example:For a sports subarticle, the system will first look for sports_uart.pbs. If this file does not exist, the system will look for sports_art.pbs. If this template does not exist, the system will look for gen_uart.pbs. The last template the system will look for is gen_art.pbs. If this template does not exist, the system will fail.

AlwaysCheckMainCatMap=1

This option introduces a rule that makes the system check main category mappings even if sub-category mappings exist, as long as the resulting file from the sub-category match did not exist. This means that if you have mapped *News* to *Sports,* News01 will use the sports template even if a specific mapping does not exist.

Other Methods

If the differences between the templates are more subtle, like color changes and different parts that are to be included, you can use SAXOTECH Online tags to include different information on different categories. These tags are described in Other Tags.

Template names and URLs

All templates in SAXOTECH Online must be named according to a predefined structure. This naming structure is based on the template type and category. All template files have the .pbs extension. All template files must be placed in the template folder under the root directory for the website.

Front page template

The front page template is called **frontpage.pbs.** This is the template that is used by the website's homepage.
The front page is called by accessing the following URL:

```
/apps/pbcs.dll/frontpage
```

You must place a default document in the root folder of the web server that redirects to this address.

Sectionfront templates

Sectionfronts are accessed through the URL:

```
/apps/pbcs.dll/section?Category=XXX.
```

Replace

```
XXX
```

with the name of the category.

Default template

The default sectionfront template is called **gen_ovr.pbs**. This template is used if no template exists for the category, or no mappings exist in the ini file.
Section templates are named using the following structure:

- The first part of the name signifies the category.
- **Gen** is the general category and is used when no other template file exists.
- The second part of the name indicates the template type. For example, **ovr** signifies that the template is an overview/section template.

Single category template

If you want to use a special template for the category News, the file should be named news_ovr.pbs. The gen portion of the name has been replaced with the category ID. For subcategories of news that are named news01....news99, the template named news_ovr.pbs will be used. This is a built-in default mapping in the system.

Sectionfront with notices

If you use the SAXOTECH Online function **Notices** on a category, the **gen_ovrn.pbs** template is used instead of gen_ovr.pbs.
Notices is a function that allows you to filter out short stories from the main section overview. Whether or not a category uses notices is defined in the category setup. To use a category template for a section where you use notices, the template should be named

```
<category>_ovrn.pbs.
```

Story templates

Default story template
Story templates are named with a structure similar to sectionfronts. The default template for stories is called **gen_art.pbs**.
To use a template for only one category, name the template:

```
<category>_art.pbs
```

As with section overviews, stories placed in subcategories named category01..99 will automatically use the same template as the main category, if no specific template exists.

Substory template

A substory is a story that has the **Substory** type selected in the **Story type** drop-down on the story properties page. Substories are not listed by the Newslist or Sectionfront tags. Substories use the template **gen_uart.pbs**. They are searchable by SAXOTECH Online Search powered by FAST, but are not searchable by the old SAXOTECH SQL server-based search.

Front page linked story

This template is used when you link to a story from the front page and have the option

```
FrontpageFlag=1
```

in the pbs newslist. The template is called **gen_fart.pbs**.

Front page linked subarticle

The **gen_fuart.pbs** is used by all subarticles listed with a main story when the

```
Frontpageflag=1
```

option is used.

Unique templates for the same story

To use a different template for the same article, create the template and then use the template name with the

```
<%link%>&template=template_name
```

option.

For example, the same story might be shown in the sections *Local* and *Sports* and its appearance should correspond to the section in which it is being viewed. To use the template **gen_sports.pbs**, enter

```
<%link%>&template=sports.
```

Other templates

Other templates are discussed with their tags.

WAP templates

This section covers the steps that have to be performed if you are going to provide services for WAP users. WAP is an acronym for Wireless Application Service and allows people to browse the web through cell phones.

Setting up WAP

The server must be set up to handle default.wml as the default document. Before you can use the WAP templates, the web server must also be set up to handle the MIME types that WAP uses. These are:

```
.wbmp image/vnd.wap.wbmp
.wml text/vnd.wap.wml
.wmlc application/vnd.wap.wmlc
.wmls text/vnd.wap.wmlscript
.wmlsc application/vnd.wap.wmlscriptc
```

Default.wml

The default.wml file must provide links to the different sections you would like to have on your WAP site. All URLs must be fully qualified. You cannot use relative paths.

Images

Images are loaded like regular images with the <img> tag, but the format must be .wbmp.

Links

All links to sectionfronts must include WAP=. This option tells the system to display the page using the WAP template. All links must be fully qualified. You cannot use relative links.

Gen_wapovr.pbs
This template is used for displaying the sectionfronts.

Gen_wapart.pbs
This template is used for viewing the story. For more information on WAP object files, please see WAPArticle.pbo.

Zones

Checking zone cookie

You can check if the zone cookie is set by accessing the front page through this URL:

```
/apps/pbcs.dll/frontpage?Checkcookie=PBCSZone
```

This will bring you to the template called **PBCSzone_setcookie.pbs**.

Chapter 6 Object Files

Object files are the pieces used for designing the layout of the output. They give you total control of how the information is presented. The object file for a tag is described with the tag.

- About Object Files
- Article classes
- Global function
- Global variables
- Inline objects
- Localdata.ini
- Object script
- Odd and Even variable picture handling
- Pbs macro
    - Election2008.pbo
- Position-based design
- System variables

Object files are the pieces that make up a story page, a Classifieds page, or any other dynamically-displayed section of your website. For example, if you were to separate a news story into its various pieces, you might have a headline, a byline, the story text, a footer, and so on. Each of these elements could be considered as an object file. You can arrange, modify and manipulate object files to display the story in different ways. You can make the headline bold and at the top of the story, or you could put it all lowercase at the bottom of the story.

This chapter describes the functionality of object files and scripting. It contains general information about how object files work. The specific information about how an object file works is placed with the tag that uses the object file.

This chapter also contains information about object script, which allows you to build conditional object files. The functionality we call Article classes, which allows the user to select a specific object file for one story, is also described in this chapter.

The use of tags and object files
It is not possible to set a variable in an object file as shown in this example:

```
<%?new_var="pbs:datecalc>"%>
```

If used in this manner, the variable new_var will be equal to <pbs:datecalc> instead of the date desired.

When you want a variable with the results of a pbs tag in it, and not the tag itself, call it as part of a macro and make the variable a global variable. In macros you can use pbs tags like `<pbs:newslist>`, `<pbs:article>,` or `<pbs:macro>` tag (nested macros). An example of this is:

```
<pbs:macro name="twodaysago" test="<pbs:datecalc dateformat=yyyymmdd days=-1>">
```

and then in the macro:

```
<%?@new_var=#test#%>
```

About Object Files

Trigger use of object files

In some tags, you must use the option **useobjects=1** to get the tag to use an object file. Other tags will automatically use the object file if it exists. This feature is documented for each individual tag.

Naming of object files

Object files for most tags are named with the following structure:

```
<Function><Objectclass>.pbo
```

Object files for the newslist tag are named with the following structure:

```
<Function><Objectclass><ArticleClass>_<Position>_<Zone>.pbo
```

Object files for the article tag are named with the following structure:

```
<Function><Objectclass><ArticleClass>_<Zone>.pbo
```

|| **Article class** | Article class is selected by the users when they are editing stories. The design options on the story properties page sets the article class for the story. Article class is a letter from A to Z. Article class must be defined in the ini file. For more information, please see [Article classes. ||

Function	Function is the name of the object file. This name is used by the tag to find the correct object file.
Object class	If you are working with a headline object file, you can have 99 different styles of headlines, numbered 1 to 99. Object class is set by the designer in the tag on the template and has a numerical value from 1 to 99.
Zone	The zone value is used to show different layouts for different geographical zones. The system only checks for a zone version if the non-zoned version is found. For example the file News2a_NOR.pbo will not be used if News2a.pbo does not exist. The zone cookie must also be set.

Directory structure

Object files are placed in a folder under the template folder called **objects**. Inside the **objects** folder, you can create a folder for each category. These folders are named after the category ID of the category, not the category name. By using the folder structure, you do not need to have all object files in the main folder. This makes it easier to keep an overview of the object files.

Folder selection

When you use object files for a tag, SAXOTECH Online looks in the **object** folder for the correct object file to use. The system will look for object files in the following order:

```
/light/<lightid>/objects/
```

This folder is used for Community Site pages. The lightid is the customer ID for the Community Site customer.

```
/objects/themes/<theme>
```

If a story has a theme, the system will look for a folder with the theme's name.

```
/objects/profiles/<profile>
```

If profiles are used, the system will look for a folder with the profile number.

```
/objects/sectioncat/<SectionCat>
```

The system looks for a folder matching the section category. The section category is the category in which the story is shown, and not the category in which the story is placed.

```
/objects/sectioncat/<MainSectionCat>
```

The system looks for a folder matching the section category. If the section category is a subcategory, the main category will be checked as well.

When using category-tree with more than two levels, two digits will be removed at a time. In the following example, five lookups will be performed.

Example 1

```
NEWS010101
NEWS0101
NEWS01
NEWS
default
```

If a site has used an odd number of digits on a subcategory, this will be regarded as the only subcategory. For example:

Example 2
(Note that there are three digits on the subcategory NEWS123)

```
NEWS123
NEWS
default
/objects/<category>
```

The system will look for a folder with the same name as the category ID.

```
/objects/<maincategory>
```

The system will look for a folder with the same name as the main category if the story is placed on a subcategory.

```
/objects/
```

The system will look in the default folder if no other match is found.

```
Global
```

The global folder is the system's fallback folder and is only used when no object files are found. This folder is maintained by SAXOTECH.

Selecting which object files to use

There are 16 possible object files that can be used. The following example illustrates which object file will be used first.

You have a newslist tag on the front page, which uses object class 1. You have multiple stories in the system, some with article classes and some without. The zone is set to ENG. When the page is built, the system will look for object files in the following way:

The newslist tag will start looking for the object file that best corresponds with the specifications on the tag. It will start somewhere in the list below and move downwards, if it does not find the corresponding file.

Newsitem1A_5.pbo	The system will use this object file if the article is in position 5, and has article class A. Position, article class and object class match.
Newsitem1A_5_ENG.pbo	The system will use this file if the previous file is found and the zone is set to ENG.
Newsitem1A.pbo	The system will use this object file if there is not an object file for the position, but there is one for the article class. Article class and object class match.
Newsitem1A_ENG.pbo	The system will use this file if the previous file is found and the zone is set to ENG.
Newsitem1_5.pbo	This file is selected if there is no object file for the article class, but there is one for the position the article has in the list. Position and object class match.
Newsitem1_5_ENG.pbo	The system will use this file if the previous file is found and the zone is set to ENG.
Newsitem1.pbo	This file is used if there is no object file for the position or article class. Only object class matches.
Newsitem1_ENG.pbo	The system will use this file if the previous file is found and the zone is set to ENG.
NewsitemA_5.pbo	This file is used if there is no match for the object class, but article class and position match.
NewsitemA_5_ENG.pbo	The system will use this file if the previous file is found and the zone is set to ENG.
NewsitemA.pbo	This file is used if there is no match for position or object class. Only article class matches.
NewsitemA_ENG.pbo	The system will use this file if the previous file is found and the zone is set to ENG
Newsitem_5.pbo	This file is used if there is no match for article class or object class. Only position matches.
Newsitem_5_ENG.pbo	The system will use this file if the previous file is found and the zone is set to ENG.
Newsitem.pbo	This file is used if there us no match for position, article class or object class.
Newsitem_ENG.pbo	The system will use this file if the previous file is found and the zone is set to ENG.

You can set up object files in subfolders (example /templates/objects/news/NewsItem.pbo). If you are using the Newslist tag on the sectionfront for the category *News,* the system will use NewsItem.pbo in *News* instead of the default NewsItem.pbo. Further, if you are using the Newslist tag on the sectionfront for the category *NEWS04,* the system will use NewsItem.pbo in *News,* instead of using NewsItem1A_2.pbo in the objects folder.

File descriptions

The 16 files are listed below in the order of increasing specification detail of the object file name.

NewsItem.pbo	Default. This file is used if the newslist tag contains no position, article class or object class.
NewsItem_5.pbo	This file is used if the newslist tag has `position=5` When the newslist tag also contains object class or article class, it will start looking for other files. You can also use the **<%Counter%>** variable in NewsItem files to match position.
NewsItemA.pbo	This file is used if the newslist tag has `articleclass=A` When the newslist tag also contains object class or position, it will start looking for other files.

NewsItemA_5.pbo	This file is used if the newslist tag has
	`articleclass=A`
	and
	`position=5`
	When the newslist tag also contains object class, it will start looking for other files. You can also use the **<%Counter%>** variable in NewsItem files to match position.

NewsItem1.pbo	This file is used if the newslist tag has `objectclass=1` When the newslist tag also contains article class or position, it will start looking for other files.

NewsItem1_5.pbo	This file is selected if the newslist tag has
	`objectclass=1`
	and
	`position=5`
	When the newslist tag also contains article class, it will start looking for other files. You can also use the **<%Counter%>** variable in NewsItem files to match position.

NewsItem1A.pbo	The system will use this object file if the newslist tag has
	`objectclass=1`
	and
	`articleclass=A`
	When the newslist tag also contains a position, it will start looking for other files.

NewsItem1A_5.pbo	The system will use this object file if the story has
	`objectclass=1`
	,
	`articleclass=A`
	and is in
	`position 5`
	You can also use the **<%Counter%>** variable in NewsItem files to match position.

If zone is set to ENG:	
NewsItem_ENG.pbo	This file is used if the newslist tag contains no position, article class or object class and the zone is set to ENG.
NewsItem_5_ENG.pbo	This file is used if the newslist tag has `position=5` and the zone is set to ENG. You can also use the **<%Counter%>** variable in NewsItem files to match position.
NewsItemA_ENG.pbo	This file is used if the newslist tag has `articleclass=A` and the zone is set to ENG.

NewsItemA_5_ENG.pbo	This file is used if the newslist tag has
	`articleclass=A`
	`position=5`
	and the zone is set to ENG. You can also use the **<%Counter%>** variable in NewsItem files to match position.

NewsItem1_ENG.pbo	This file is used if the newslist tag has `objectclass=1` and the zone is set to ENG.

NewsItem1_5_ENG.pbo	This file is selected if the newslist tag has
	`objectclass=1`
	`position=5`
	and the zone is set to ENG. You can also use the **<%Counter%>** variable in NewsItem files to match position.

NewsItem1A_ENG.pbo	The system will use this object file if the newslist tag has
	`objectclass=1`
	`articleclass=A`
	and the zone is set to ENG.

NewsItem1A_5_ENG.pbo	The system will use this object file if the story has
	`objectclass=1`
	`articleclass=A`
	is in
	`position 5`
	and the zone is set to ENG. You can also use the **<%Counter%>** variable in NewsItem files to match position.

Article classes

Article classes allow you to predefine multiple templates that the users can select when they write stories. Before the user can do this, the following must be set up:

Setup
The following code must be present in the Publicus.ini file for the site.

FRONTPAGESUBCLASS section

```
a=Breaking news
b=Major event
```

SECTIONSUBCLASS section

```
a=Breaking news
b=Major event
```

ARTSUBCLASS section

```
a=Breaking news
b=Major event
```

Templates

Article	The article template must be built using object files.
Front page	The front page must be built using the pbs newslist. This is the only overview tag that supports article classes.
Sectionfront	The sectionfront must be built using the pbs newslist. This is the only overview tag that supports article classes.

Object files

The naming of the object files for use with article classes is as follows.

Article	For the object files used by the article tag, you add the article class letter to the end of the name. The object file for the heading will be named **ArticleHeadingA.pbo**. For object class 2 and article class B, the file must be named **ArticleHeading2B.pbo**.
Newslist	The pbs newslist tag must be used on both the front page and the sectionfront.The object files used by the newslist tag are called **News.pbo** and **NewsItem.pbo**. When you use an object class, the object class number is input before the article subclass and the files are named **NewsItem2A.pbo** and **NewsItem2B.pbo**. The object files can also be placed in a sub folder with the same name as the category. The file **News2.pbo** is used for all stories.

Example
A system has been set up with two subclasses for the front page. These are called *Breaking News* and *Major Event*. *Breaking News* is used when something happens during the day, and *Major Event* is used when something major happens, like earthquakes, assassinations, and so on. Stories have been written using all the different subclasses. To present the information in the database, the following tags and object fields are used.

Tag

The following newslist tag is used to build the main page.
<pbs:newslist categories="news%,sports%" count=15 days=2 priosort useobjects=1 Objectclass=1>

Object files

News1.pbo

The main file that is used for the entire list.

```
<font face="<pbs:fontface1>" size=1>
<%items%>
</font>
```

NewsItem1.pbo

Used for the regular items.

```
<br clear=all>
[%{Imagelink}<img src="<%Imagelink%>&MaxW=100"
width=100 border=0 hspace=1 vspace=1 align="right">%]
<font face=Verdana size=2><b><%\Title%></b></font><br>
<font face=Verdana size=1><%Leadin%></font><br>
<div align="left"><font face=Verdana size=1>
<a href="<%Link%>">Whole story</a> |
<a href="<%pbcs.dll%>/section?Category=<%Category%>">
<%Categoryname%></a></font></div>
<br>
</font>
```

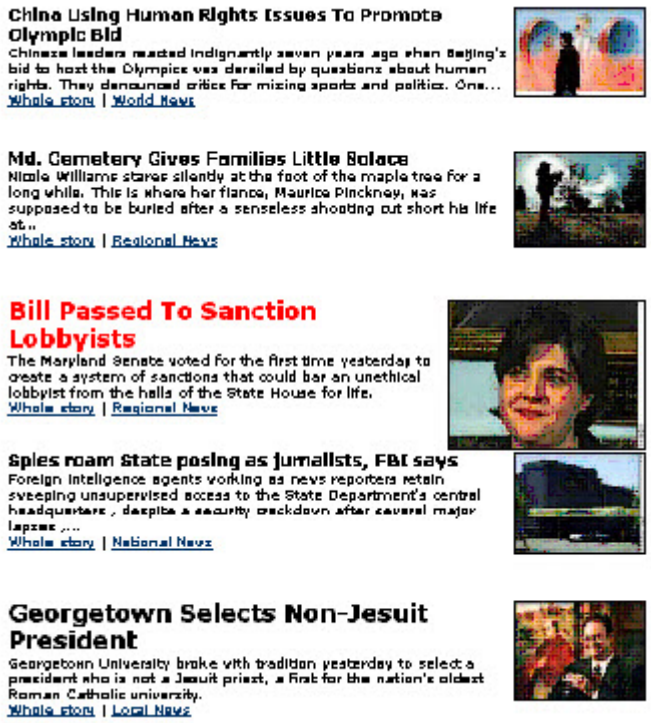
NewsItem1A.pbo

Used for all items marked as *Breaking News.*

```
<br clear=all>
[%{Imagelink}<img src="<%Imagelink%>&MaxW=100"
width=100 border=0 hspace=1 vspace=1 align="right">%]
<font face=Verdana size=4>
<b><%\Title%></b></font><br>
<font face=Verdana size=1><%Leadin%></font><br>
<div align="left"><font face=Verdana size=1>
<a href="<%Link%>">Whole story</a> |
<a href="<%pbcs.dll%>/section?Category=<%Category%>">
<%Categoryname%></a></font></div>
<br>
</font>
```

NewsItem1B.pbo	Used for all items marked as *Major Event.* `<br clear=all>` `[%{Imagelink}<img src="<%Imagelink%>&MaxW=150"` `width=150 border=0 hspace=1 vspace=1 align="right">%]` `<font face=Verdana size=4 color="#FF0000">` `<b><%\Title%></b></font> ` `<font face=Verdana size=1><%Leadin%></font> ` `<div align="left"><font face=Verdana size=1>` `<a href="<%Link%>">Whole story</a> |` `<a href="<%pbcs.dll%>/section?Category=<%Category%>">` `<%Categoryname%></a></font></div>` ` ` `</font>`

Example page
With A and B class stories:


China Using Human Rights Issues To Promote Olympic Bid
Chinese leaders reacted indignantly seven years ago when Beijing's bid to host the Olympics was derailed by questions about human rights. They denounced critics for mixing sports and politics. One...
Whole story | World News
Md. Cemetery Gives Families Little Solace
Nicole Williams stares silently at the foot of the maple tree for a long while. This is where her fiance, Maurice Pinckney, was supposed to be buried after a senseless shooting cut short his life at...
Whole story | Regional News
Bill Passed To Sanction Lobbyists
The Maryland Senate voted for the first time yesterday to create a system of sanctions that could bar an unethical lobbyist from the halls of the State House for life.
Whole story | Regional News
Spies roam State posing as journalists, FBI says
Foreign intelligence agents working as news reporters retain sweeping unsupervised access to the State Department's central headquarters, despite a security crackdown after several major lapses, ...
Whole story | National News
Georgetown Selects Non-Jesuit President
Georgetown University broke with tradition yesterday to select a president who is not a Jesuit priest, a first for the nation's oldest Roman Catholic university.
Whole story | Local News

Global function

The global function is used to output the global and system variables in a template. User defined global variables and system variables can be used. The function uses an inline object like this:

```
<pbs:#global# object="<%currentdate$d(\"dddd, mmm. d\")%>">
```

Global variables

Global variables should be used sparingly! They can significantly impact performance on your website.

The following conditions apply to global variables:

- On a template, a global variables must be set in a macro. There must be a cache time on the macro.
- If you use an additional macro on the same template, which refers to the global variable, the cache time MUST be the same as on the initial global variable macro.
- If you have a newslist tag on the same template, the macro on the news item will NOT have access to the global variable, even if the newslist tag has the same cache time as the global variable macro.
- Global variables only work on a single level and only when ALL cache times are the same.

Global variables are available in all object files. These variables are prefixed with ?@ like this:

```
<%?@hasimages%>
```

You can write global variables using object script.

Testing

You can use the following code to test if global variables are available in the object file:

```
[%{GlobalVariablesSupported=1}This objectfile supports global variables.%]
```

You can use the global variables in the same way as all other variables.

To set the global variable, use this syntax:

```
<%?@bar="value to be used"%>
```

Examples

Macro1.pbo	Sets the variable @bar: `<%?@bar= "12"%>`
Macro2.pbo	Activates the variable @bar: `<%@bar%>`
Call both macros from a PBS-file	`<pbs:name= "Macro1"> <pbs:name= "Macro2">`

Inline objects

Inline objects allow you to format the string output with `<pbs:#variable#>` tags. The object is formatted like this:

```
<pbs:#ureg_message# object="[%<%variable$r(\"Wrong=Incorrect\")%>%]">
```

The quotation marks must be preceded by a backslash for the function to work.

Example

```
<a href="<pbs:!#LastReferrer# object="[%<%Variable%>%|%javascript:history.back();%]">">
```

Inline objects can also be used as a macro to run, for example, a simple check. If used in this way, it is not necessary to have a macro file in the macros folder.

Example

```
<pbs:macro inlineobject="[%{category}Yes%|%No%]">
```

Localdata.ini

Localdata.ini is a file used to hold your own permanent global variables. This is useful for storing values that you use in multiple places on your site. The file has the following sections:

Category keys	Key lookup is used to look up values based on variables:
	`[Category_Keys]`
	`Sports=Ball sports`
	`Sports01=Footbal`
	`Sports(\d)=Sport$1`
	`News=Bar`
	This allows you to perform a lookup on a field and replace the value of the field with the value from the key lookup table. In the example above, a field called *category* with the value *Sports* will show *Ball sports*. If the field has the value sports01, the value returned will also be *Ball sports*. The reason for this is that the field is matched in the order they are listed. On the first match found, the system will stop looking. For this reason you must put the most generic match option at the bottom of the list.
	Options
	`$k`
	means exact match. The match is case insensitive.
	`$K`
	means regular expression. The regular expression follows the standard rules for regular expressions.

STATICVALUES	`[STATICVALUES]`
	`MaxImageWidth=200`
	`NoRecordsFoundMessage=No Customers Found`
	This will allow you to use the variable
	`<%MaxImageWidth%>`
	in all macros.

Object script

Object script is the functionality that allows you to make conditional decisions in object files based on the content of the fields in the object file. In addition, object script allows you to do arithmetic with the values of numerical variables.

Test if a variable has content

If a piece of code is only to be output if the variable contains information, you can enclose the whole line or section in `[% ... %]` tags.

In the following example, the break after the field will only be output when

```
<%text%>
```

has content:

```
[% <%text%> <br> %]
```

Multiple variables

If you have more than one field in the block, you must write the test like this:

```
[%{Link}<a href="<%Link%>"><%Title%></a>%]
```

With two or more variables, you must specify which variable you want to check. This is done by enclosing the variable in {}.

Example
The following piece of code is only output when the variable Imagelink is not blank:

```
[%{Imagelink}
  <table border="0" cellspacing="0" cellpadding="2" align="right" bgcolor="#EEEEEE">
    <tr><td><img src="<%Imagelink%>&MaxW=200&MaxH=150&Q=85" border=0></td></tr>
    <tr><td align="left"><%Caption%></td></tr>
  </table>
%]
```

Reverse test

You can also use this as a reverse test using ! which is the operator NOT:

```
[%{!Link}There is no link%]
```

In this example, the text will be output when there is no link.

Nesting tests

You can nest tests inside each other. In this example, the table row with the caption will only be output if the caption exists:

```
[%{Imagelink}<table border="0" align="right" bgcolor="#EEEEEE">
<tr><td><img src="<%Imagelink%>&MaxW=200&MaxH=150&Q=85" border=0></td></tr>
[%<tr><td align="left"><%Caption%></td></tr>%]
</table>
%]
```

Alternate output

You can also output different text based on the value of a variable. The format looks like this:

```
[%{variable}The variable has a value %|% The variable has no value %]
```

Example
In this example, the title will be a link only when the link field has a value:

```
[%{Link}<a href="<%Link%>"><%Title%></a>%|%<%Title%>%]
```

Value tests

It also possible to check against the value of a field. This allows you to do a more advanced test than merely checking if a field has a value.

Basic math

You can preform the following tests:

```
{Width>240} which is true when <%Width%> is larger than 240.
  {Width<240} which is true when <%Width%> is smaller than 240.
  {Width=240} which is true when <%Width%> equals 240.
  {Width!240} which is true when <%Width%> does not equal 240.
```

Example
In the following example, the field *Width* is checked to see if it contains a value that is lager than 240. Code is output based on the check:

```
[%{Width<240}
  <img src="/g/t.gif" width="240">
%|%
  <img src="/g/t.gif" width="<%Width%>">
%]
```

The width of the image will be at least 240 pixels. If the width field has a higher value, the image will retain its value.

Odd or even test

This function allows you to build a newslist in which you can alternate pictures on the left or right. For more information, please see Odd and even.
Use the following test to check if the variable is odd:

```
[%{Width is odd}Output this text when width is odd%]
```

Use the following test to check if the variable is even:

```
[%{Width is even}Output this text when width is even%]
```

Advanced value test

You can also compare the numerical value of two variables. This expands on the basic test that compares the variable to a number. When you use this function, the second variable must be enclosed in hash signs #. The test must be placed inside curly brackets {}.

Example 1
In this example, the last changed date is only output if the story has been modified after it was written. You must use the ISO versions of the variables when you define a date test.

```
Published: <%PublishedDate%>
[%{ISOModifiedDate>#ISOPublishedDate#} - Last changed: <%ModifiedDate%> %]
```

Example 2
In this example, the position of the picture is adjusted, depending on whether the picture is higher than it is wide. If the photo is wider than it is high, the photo will be aligned left. If the photo is higher than it is wide, it will be aligned right. The photo will also be scaled to a maximum width of 200 pixels or a maximum height of 300 pixels. This function is built into the image function:

```
<img src="<%ImgLink%>&MaxH=300MaxW=200" Align="[%{Width>#Height#}Left%|%Right%]" />
```

Multiple tests

It is also possible to perform multiple tests to determine if the code should be output. This function uses an ampersand **&** for **AND**. It uses a pipe **|** for **OR**. & has a higher priority than |.

Example

```
[%{Category=INTERNET&ImageLink|Category=NEWS}<font size=4>%|%<font size=2>%]
```

In this example, the font size will be set to 4 if one of the following statements is true:

- The category is *Internet* **and** ImageLink contains a value (not blank).
- The category is *News.*

Formatting dates

To use date formatting, you must use the operator **d**. The options below are available for this operator.

If available, try using ISO versions of date and time variables:

```
<%isodate%>
```

```
<%isodatetime%>
```

and so on. The <%isodate%> of an evergreen article will always be 99999999hhmm, where hhmm is the site's rollover time. If you do not want the hhmm on the date, use **$c(1,8)** instead of date formatting. This will return the IsoDate for most articles and 99999999 for evergreen articles.

Please check available variables for the object file in question.

am/pm	Outputs the time in 12 hour format.
c	Date in a predefined short version and time in a predefined long version.
d	The number of the day of the month, without leading zero.
dd	The number of the day of the month, with leading zero.
ddd	Short name of the day (Mon, Tue, Wed, etc.).
dddd	Long name of the day (Monday, Tuesday, etc.).
ddddd	Date in a predefined short version.
dddddd	Date in a predefined long version.
h	Hour without leading zero.
hh	Hour with leading zero.
m	Month number without leading zero.
mm	Month number with leading zero. mm can also mean minute if it is preceded by hh.
mmm	Short name of the month (Jan, Feb, Mar, etc.).
mmmm	Long name of the month (January, February, etc.).
n	Minute without leading zero. M is used for month, so N is used instead.
nn	Minute with leading zero.
s	Second without leading zero.
ss	Second with leading zero.
t	Time in a predefined short version.
tt	Time in a predefined long version.
yy	Year with two numbers. For example 07.
yyyy	Year with four numbers. For example 2007.

Example
You have the variable <%isodate%> in your object file and you would like to use a format other than the default format. To change the date's format, you could write:

```
<%isodate$d("mmmm d, yyyy")%>
```

If the value for <%isodate%> is 20010601, this would result in the date being written like this:
June 1, 2001

Formatting text

You may want to use a different formatting of a field, rather than the default supported by the variable. To limit the number of variables the object files must support, it is possible to specify the formatting directly in the pbo file. You cannot use these functions on fields that

include other object files.

Syntax

The syntax is as follows:

```
<%VarName$<operator>("<format>")%>
```

It should be noted, that it is possible to have several $ operators on a variable. These are executed from left to right.

$	The delimiter that tells the system that formatting for the variable is supplied.
<format>	This field is used to format the variable.
<operator>	Tells the system which class of formatting should be used. The operator is case sensitive. Only the formatting options are listed here. There are other operators available for extracting and replacing text. Available operators are: • **$e** is used to replace ASCII characters 161 to 255 with HTML entities. • **$j** is used to convert text from plain text to URL encoded text. • **$J** is used to convert text from URL encoded to plain text. • **$h** is used to convert the variable from text to HTML. • **$H** is used to convert the variable from HTML to text. • **$l** is used to convert the variable to lowercase. For example, <%Superheading$l%> will show as 'this is a test' • **$n** is used to convert the first letter of every word to uppercase (Name case). For example, <%Superheading$n%> will show as 'This Is A Test' • **$u** is used to convert the variable to uppercase. For example, <%Superheading$u%> will show as 'THIS IS A TEST' • **$p** is used to convert the variable to Norwegian proper case. For example, <%Superheading$p%> will show as 'This is a test' • **$b** is used to convert the variable to US proper case. • **$t** is used to strip out all HTML codes. • **$o** converts date to text in FAST InStream search compatible format. • **$O** converts datetime to a text in FAST InStream search compatible format. • **$C** Surrounds variable contents in CDATA tags. Use (F) to force the tags if no special characters are present. For example: <%Title$C(F)%> • **$s** Returns the length of the variable • **$D** Converts ISODateTime into Extended format. For example: <%ISODateTime$D%> will output in a GMT minus local time offset, like this *Wed, 13 Feb 2008 16:16:00 -0400* • **$S** Runs a script, which name is given as the input parameter. The second paramter given is the script language. The second parameter can be left out if the script has a .vbs extension or by adding the file extension to the name. For example: <%currentdomain$S("myScript")%> will call the script myScript.vbs. Both <%currentdomain$S("myScript","JScript")%> and <%currentdomain$S("myScript.js")%> will call the script myScript.js. The script must be places in the scripts folder of the templates directory. The name and value of the variable the script is executed on (in the previous example, this is currentdomain) is avaliable through SOSE.GetVar("_Variable") and SOSE.GetVar("_VariableValue")
VarName	The name of the variable.

Formatting numbers

This function is used to format numbers output by the system. You can select the desired number of decimals, separators, and so on. The function is triggered by $f, like this:

```
<%Amount$f("#,##0.00")%>
```

The following table shows how the format affects different values.

Format	1234	-1234	0.5	0
0	1234	-1234	1	0
0.00	1234.00	-1234.00	0.50	0.00
#.##	1234	-1234	.5	
#,##0.00	1,234.00	-1,234.00	0.50	0.00
#,##0.00;(#,##0.00)	1,234.00	(1,234.00)	0.50	0.00
#,##0.00;;Zero	1,234.00	-1,234.00	0.50	Zero

0.000E+00	1.234E+03	-1,234.00E+03	5.000E-01	0.000E+00
#.###E-0	1.234E3	-1.234E3	5E-1	0E0

Example

```
<td>Price:</td><td>$<%Amount$f("#,##0.00")%></td>
```

The system will output the correct thousand and decimal separator based on the local setting on your FTP server.

Multiple formatting operations

You can also use multiple formatting operations on the same variable.
<%Date$d="dd. mmmm yyyy"%> will output "16. May 2001".

Output the month in lowercase by combining d and l (lowercase) like this:
<%Date$d("dd. mmmm yyyy")l%> you will get "16. may 2001".

Notice that l must be last, because the commands are executed in the order they are listed. The commands must be placed in front of the formatting information (="dd. mmmm yyyy") and only one command that requires formatting can be used.

Arithmetic

You can add, subtract, multiply, divide and perform the modulo operation with the values in variables. This is only possible for the variables that contain an integer. An integer is a number without decimals.

Syntax

```
<%VarName#[+,-,*,/,%]<value>%>
```

If you do not specify what kind of operation you are performing, the number will be added.

Varname	The name of the variable.
#	Tells the system to perform an operation on the value.
+	Tells the system to add the value to the variable.
-	Tells the system to subtract the value from the variable.
*****	Tells the system to multiply the value with the variable.
/	Tells the system to divide the variable by the value.
****	Used to check if a variable is divisible without leaving a fraction. If this is true, the value will be true. You use it like this: `[%{Counter\4}This text is output if counter is divisible by 4 without a fraction%]`

%	The % tells the system to perform the modulo operation, which is the operation of finding the remainder in division. For example, 10 modulo 4 = 2 because when you divide 10 by 4, 2 is the remainder.You can also use the modulo operation to output different code based on the remainder. Entering {`[Counter%3]=1`} will be true when the counter has the value 1, 4, 7 etc. This allows you to make a three column output like this:
	`[%{[Counter%3]=1}<tr>%]`
	`<td>Content</td>`
	`[%{[Counter%3]=0}</tr>%]`
	The table row start and end will only be output at the start and end of the line.

<value>	This is the number you would like to use in the operation.

Examples
In this example, the field `<%Width%>` contains the number 240:

```
<%Width#8%> = 248
<%Width#+8%> = 248
<%Width#-8%> = 232
<%Width#*2%> = 480
<%Width#/2%> = 120
<%Width#%2%> = 0 (modulo is the remainder of a division)
```

Arithmetic and formatting

You can combine arithmetic and formatting of variables. The calculation must be done first.

<%CashFlow#/2$f("#,##0.00")%> will apply thousands separators after dividing the value of the variable by two. If <%CashFlow%> contained the value "2432", the result will be "1,216.00".

Variable check

You can check if a variable contains a text value. The match looks at the whole variable. You cannot use this function to match a substring. You can check for a substring using a regular expression.

Case insensitive	{{[% Unknown macro: {Name=Bob} %]}} will output something if the name is Bob. This check is case insensitive.
Case sensitive	By typing {{[% Unknown macro: {Name==Bob} %]}} the check is case sensitive.
Quote marks	If you are going to check a string that contains quotation marks, you must prefix the quotation marks with back slashes like this: {{[% Unknown macro: {Name=""Fast" Willy Parker"} %]}}
Special characters	You can check for values containing spaces or the special characters & and | by using quotation marks around the string like this: {{[% Unknown macro: {Department="Research & Development"} %]}}

Extracting a substring

This function allows you to extract a substring of a variable. There are three different substring functions.

Character	Use the option like this:
	`$c(start,count)`
	Start is the character with which you start, count is the number of character you would like to show. If you leave the count blank, it will show the rest of the string. For example:
	`<font size="5"><%Heading$c(1,1)%></font><%Heading$c(2)%>`
	This example will make the first letter in the heading bigger.

Line	Extracts lines from a multiline text. Use the option like this:
	`$L(LineStart,LineStop)`
	LineStart is the line with which you start. LineStop is the line with which you stop.
	`$l`
	converts the text to lowercase.

Word	Copies a substring and then truncates the last character or word in the text to ensure it ends with a whole word. Use the option like this:
	`$w[(StartPos[,count])]`
	Startpos is the character with which you start, count is the number of characters you would like to show. If you leave it blank, it will show the rest of the string. This function does the same as the
	`$c`
	option, but it does not break words.

Length check

You can also check the length of a text string. The basic test is to check the value of a string. By adding [ ] (square brackets) around the field, you can check the length instead.

Example:

You would like to shorten titles so they stay on one line on the front page. In addition, you would like to add ... to show that the title is longer. If the title is shorter than 50 characters, the whole title will be shown without the dots.

```
[%{[Title]>50}<%Title$c(1,47)%>...%|%<%Title%>%]
```

This test will cut titles longer than 50 characters at 47 characters and add three dots.

URL replace

This function allows you to make internet addresses in text fields into hyperlinks. The function will automatically convert all addresses starting with http:// ftp:// and www. into hyperlinks. All e-mail addresses will be converted into `mailto:` links.

To use this function, add $a to the field on which you would like to use the function. You cannot use this function on a field that calls another object file.The body field in ArticleParagraph.pbo looks like this with the function added:
<%Body$a%>

Options

You can set the target and classes for the links. This is done by adding the following options:
$a(target,urlclass,mailclass)

All parameters are optional, so you can write:
$a, $a(_top), $a(_top,extlink,maillink) or $a(,extlink).

Replace text

This function allows you to replace text in a variable. The function is case insensitive.

Example 1	You can use the function like this:<%Body$r("Banana=Apple")$r("Orange=Lemon")%>This function will replace every occurrence of the word banana with apple and every occurrence of the word orange with lemon.
Example 2	You can also use the function to change the way a word is written:<%Body$r("saxotech=SAXOTECH")%>This function will ensure that SAXOTECH is always written in uppercase letters.

Example 3	The quotation mark is a special character and must be doubled up for the system to understand it. If you want to replace a double quotation with a single quotation you can write it like this:<%Body$r("""="")%>
Example 4	The backslash is a special character and must be doubled up for the system to understand it. If you want to replace a double backslash with a single backslash you can write it like this:<%Body$r("\=backslash")%>Backslash is used for specifying regular expressions. Please see the **Regular expresions** section below.

Regular expressions

Regular expressions can be used in the replace function. This expands the basic replace function.

Finding documentation	You can find the documentation of **regexp** on the web by searching for the phrase "perl regexp". There are also books available on this subject.
Matching text	There are several functions for matching text: • **.** (period) Matches any single character except a new line character. • * Matches the preceding character zero or more times. • **+** Matches preceding character one or more times. • **^** Matches the beginning of the line. • **$** Matches the end of line. • **[xyz]** Matches a character set. Matches any characters between brackets. • [^xyz] Matches a negative character set. Matches any characters NOT between brackets. • \| Used for creating OR statements (expression) Brackets an expression to use in the replace command. A regular expression may have up to 9 tagged expressions, numbered according to their order in the regular expression. • * Marks the next character as a special character. "w" matches the character "w". "\w" matches any character. The opposite is also true. • **$** Matches the end of the line "$" matches the dollar sign. • **\w** Matches one alphanumeric character. • **\W** Matches any non alphanumeric character. • **\d** Matches one digit. • **\t** Matches a tabulator. • **\r** Matches carriage return. • **\n** Matches new line. • **\s** Matches any white space character. For example: space, tabulator, carriage return and new line.
. (period)	Matches any single character except a new line character.
*****	Matches the preceding character zero or more times.
+	Matches preceding character one or more times.
^	Matches the beginning of the line.
$	Matches the end of line.
[xyz]	Matches a character set. Matches any characters between brackets.
^xyz	Matches a negative character set. Matches any characters NOT between brackets.
| (pipe)	Used for creating OR statements
(expression)	Brackets an expression to use in the replace command. A regular expression may have up to 9 tagged expressions, numbered according to their order in the regular expression.
****	Marks the next character as a special character. "w" matches the character "w". "\w" matches any character. The opposite is also true. "$" matches the end of the line "/$" matches the dollar sign.
\w	Matches one alphanumeric character.
\W	Matches any non alphanumeric character.
\d	Matches one digit.
\t	Matches a tabulator.
\r	Matches carriage return.
\n	Matches new line.
\s	Matches any white space character. For example: space, tabulator, carriage return and new line.
. (period)	Matches any single character except a new line character.
*****	Matches the preceding character zero or more times.

+	Matches preceding character one or more times.
^	Matches the beginning of the line.
$	Matches the end of line.
[xyz]	Matches a character set. Matches any characters between brackets.
[^xyz]	Matches a negative character set. Matches any characters NOT between brackets.
*****	Used for creating OR statements
(expression)	Brackets an expression to use in the replace command. A regular expression may have up to 9 tagged expressions, numbered according to their order in the regular expression.
****	Marks the next character as a special character. "w" matches the character "w". "\w" matches any character. The opposite is also true. $ matches the end of the line "\$" matches the dollar sign.
\w	Matches one alphanumeric character.
\W	Matches any non alphanumeric character.
\d	Matches one digit.
\t	Matches a tabulator.
\r	Matches carriage return.
\n	Matches new line.
\s	Matches any white space character. For example: space, tabulator, carriage return and new line.
Special characters	SAXOTECH Online uses = (the equal sign) as a separator. If you want to search for =, you must use the code **\x3D** for =. Quotation marks must doubled like this: ÒÒ, because the first one signifies that the next character is a special character.
Syntax	`$r` triggers the use of regular expressions:<%Body$r("Bob=Robert")%>
Turning off case insensitive	The options "case insensitive" and "greedy" are turned on by default. You can turn these functions off by writing `(?-i)` and `(?-g)` at the beginning of the expression.
Variables	The following variables are available:$0 contains the whole matching string.$1, $2 etc, will contain the parts if the expression is split up with parenthesis. In the function $r("(\w+)Orange(\w+)=$1 Lemon $2", which is a more complex way of replacing text, $0 contains the whole variable, $1 contains the text in front of Orange and $2 contains the text after Orange.

Examples

Matching text

This matches one or more alphanumeric characters:

```
\w+
```

This matches Smith or Jones:

```
Smith\|Jones
```

This matches the entire string:

```
^.+$
```

This matches "man", "men", "min" but not "moon":

```
m.n
```

This matches "test", "teest", "teeeest" etc. It does not match "tst":

```
Te+st
```

This matches "test", "teest", "teeeest" etc. AND "tst":

```
Te*st
```

Adding text

Use this function to add text. In the example the string "Alice, Bob and Charlie went on a road trip" is changed to "cool Alice, cool Bob and Charlie went on a road trip":

```
$r("(Alice | Bob)=cool $0")
```

When the system finds Alice or Bob, it will replace the word with the text on the right hand side.

```
$0
```

is the matching text.

Changing text

The quotation mark is a special character and must be doubled up for the system to understand it. If you want to replace a double quote with a single quote you can write it like this:

```
<%Body$r("""="")%>
```

You can also change text in the string. There are multiple ways to do this:

Example a
The matching word is replaced with another word.

```
<%Body$r("Bob=Robert")%>
```

Example b
This code will change "We see that x=4" to "We see that x equals 4". Here we are looking for the = sign. Since this is a special character we must look for the hex value instead. In addition to replacing the equal sign, a space is inserted before and after the text.

```
<%Body$r("\x3D= equals ")%>
```

Example c

```
<%Body$r("(\w)\x3D(\w)=$1 equals $2")%>
```

The equal sign will only be replaced if an alphanumeric character is next to the equal sign on both sides.

```
$0
```

contains "x=4" in this example.

Removing text

To remove the text (unknown address) from the story byline you can use this function:

```
<%BylineEmailPart$r("(unknown address)=")%>
```

The text matched on the left hand side is not replaced, so the text is removed from the string. The backslashes in front of the parenthesis are there so they are treated as characters and not operators.

Extracting text

You can also extract a part of a string.

```
<%currentrequest$r("(^.+category\x3d)(\w+\d*)(.*)=$2")%>
```

The first parenthesis looks for text from the beginning of the line to category=. The second parenthesis looks for x amount of letters and zero or more numbers. The third parenthesis looks for the rest of the string. The second parenthesis will contain the category ID because it will find all text and numbers between = and &.The function as used here complicates the problem. <%Currentrequest%> is only available in a macro, which splits all URL parameters into their own fields. The field <%category%> should be used instead, since it contains the same information.

Wrapping text

You can also match the entire field to output text before or after the field. Since . does not match new line, every line in the field will be wrapped in paragraph tags.

```
<%Body$r("(?-g)^.+$=<p>$0</p>")%>
```

This function will put every line output inside \ . "Greedy" has been turned off so each line is wrapped instead of the whole body field.

Substring match

This function can be used for checking if a field contains a text string and outputting the text "Order print" if it does.

```
<%photo$r("(^.+)staff(.+)=1")V(sh_staffphoto)%>
[%{sh_staffphoto=1}Order print%]
```

The function sets a flag if the text "staff" is in the photo credit field. There is then a test to see if the flag is set to output the text. A temporary variable is used for this. Please see Writing new variables below for more information on temporary variables.

Extended replace

Extended replace is another replace function. The difference between regular expressions and extended replace are subtle. The big difference is that the extended replace function can use variables inside the expression.

Syntax

$x triggers the use of regular expressions:

```
<%Body$x("OldText","NewText"[,"ReplaceFlags"])%>
```

The Replace flags parameter is optional.

Replace flags

Replace flags are used to control how the replace function should behave. The flags are switched **on** using + and **off** using using - in the text string, written as the third parameter in the command. Every flag must have either a + or - specified immediately after the flag itself. For example:

```
<%Body$x("OldText","NewText","r+i+g-m+")%>
```

These options will turn regular expression mode on, ignore case on, greedy mode off and multi line mode on.

Regular expression flags

These extra flags are available when you are using regular expressions.

- **g**: Greedy mode. On by default.
- **m**: Multi line mode. Off by default. When this function is turned on, the ^ symbol will match every line beginning and $ will match every line end. When this function is turned off, the ^ symbol will match the beginning of the text and $ will match the end of text.
- **s**: Single line mode. Off by default. When this function is turned on, . (period) will match any character including ^ and $. When this function is turned off, . (period) will match any character except ^ and $.

Standard flags

- **i**: Ignore case. Off by default.
- **a**: Replace all occurrences. On by default. This flag has no effect when using regular expressions. When using regular expressions, all occurrences will always be replaced.

- **r**: Use regular expression. Off by default. If "r+" is included in ReplaceFlags, the "NewText" is treated as a regular expression substitution template. This means $0 will return the expression match and $1..$n will return the sub expression matches.

Special characters

In both OldText and NewText, quotation marks (") must be escaped like this: \". And consequently backslashes must be escaped like this:

```
```

If these special characters are not escaped, the function will not work.

Variables

Any variable value can be used in OldText and NewText. This is done by joining static text and a #Variable# expression with '+', or using #Variable# expressions alone.If you have the summary in the variable <%summary%> with the value "The writer believes this is wrong." and you have the byline in the variable <%Byline%> with the value "Bob Toner", then the expression

```
<%summary$x("The writer", "#Byline#")%>
```

will output "Bob Toner believes this is wrong".

Examples

Adding text

This will find all occurrences of "saxotech" in <%Body%>, regardless of case setting, and replace them with "SAXOTECH" in yellow text.

```
<%Body$x("saxotech", "<font color=\"Red\">SAXOTECH</font>", "i+")%>
```

Adding to a variable*

This function adds on the value of counter when the counter value is found in the text.

```
<%number$x(#Counter#+": ", #[Counter+1]#+": ")%>
```

Adding variable

If the <%\Title%> field has the value "The team wins again and again" and the <%Team%> variable has the value "BMW Williams", the function will output "The BMW Williams team wins again and again".

```
<%Title$x("The team", "The "#Team#" team")%>
```

Multiple replaces*

If you want to use multiple replace function on the same variable, you can write multiple replace commands.

```
<%field$x("apples", "oranges")x("cars", "trucks")%>
```

Regular expression

This will put every line output of <%Body%> inside and .

```
<%Body$x("^.+$", "<p>$0</p>", "r+m+g-")%>
```

Replace with variable

The function replaces the text yours truly with the value from the Name variable.

```
<%Body$x("yours truly", #Name#)%>
```

Replacing text

This will find all occurrences of "apples" in <%Body%> and replace them with "oranges".

```
<%Body$x("apples", "oranges")%>
```

Writing new variables

You can write the result of a function to a new variable using the V option in the substring match example:

```
<%photo$r("(^.+)staff(.+)=1")V(sh_staffphoto)%>
```

Use a new variable called

```
sh_staffphoto
```

It is possible to overwrite system variables. Therefore, all your variables should be prefixed with **sitecode**_. If you overwrite system variables, there might be severe consequences.

Local variables

To create a local variable, prefix with ?:

```
<%?sh_myvariable%>
```

This variable is only available in the current macro. For example:

To set a local variable, "sh_foo" with the value "10":

```
<%?sh_foo=10%>
```

To activate this variable:

```
<%sh_foo%>
```

It is possible to overwrite system variables. Therefore, all your variables should be prefixed with **sitecode**_. If you overwrite system variables, there might be severe consequences.

Global variables

To create a global variable, prefix with ?@:

```
<%?@sh_myvariable%>
```

The value is also available for other macros used on the page. For more information, please see Global variables. For example:

To set a global variable, "@sh_bar" with the value "12":

```
<%?@sh_bar=12%>
```

To activate this variable:

```
<%@sh_bar%>
```

It is possible to overwrite system variables. Therefore, all your variables should be prefixed with **sitecode**_. If you overwrite system variables, there might be severe consequences.

Odd and Even variable picture handling

You can use the counter variable to alter the layout of photos in the list of stories.

In a table

Example code

```
[%{Imagelink}

<table align="%][%{Counter is even&ImageLink}right%][%{Counter is
odd&ImageLink}left%][%{Imagelink}" bgcolor="#CCCCCC" cellspacing="0" cellpadding="2" border=0>
<tr><td><img src="<%Imagelink%>&MaxW=100" width=100 border=0 hspace=0 vspace=0></td></tr>
</table>%]
<font face=Verdana size=2><b><%Title%></b></font><br>
<font face=Verdana size=1><%Leadin%></font><br>
<div align="left"><font face=Verdana size=1>
<a href="<%Link%>">Whole story</a> |
<a href="<%pbcs.dll%>/section?Category=<%Category%>">
<%Categoryname%></a></font></div>
<br>
</font>
```

Example news list

China Using wRights Issues To Promote Olympic Bid
Chinese leaders reacted indignantly seven years ago when Beijing's bid to host the Olympics was derailed by questions about human rights. They denounced critics for mixing sports and politics. One...
Whole story | World News
Md. Cemetery Gives Families Little Solace
Nicole Williams stares silently at the foot of the maple tree for a long while. This is where her fiance, Maurice Pinckney, was supposed to be buried after a senseless shooting cut short his life at...
Whole story | Regional News
Bill Passed To Sanction Lobbyists
The Maryland Senate voted for the first time yesterday to create a system of sanctions that could bar an unethical lobbyist from the halls of the State House for life.
Whole story | Regional News
Spies roam State posing as Journalists, FBI says
Foreign intelligence agents working as news reporters retain sweeping unsupervised access to the State Department's central headquarters , despite a security crackdown after several major lapses ....
Whole story | National News
Georgetown Selects Non-Jesuit President
Georgetown University broke with tradition yesterday to select a president who is not a Jesuit priest, a first for the nation's oldest Roman Catholic university.
Whole story | Local News

Topstory Variable

The Topstory variable is set to 1 when the priority for the story is from 1 to 3. You can use this variable to set the size of the title.

NewsItem.pbo	With this NewsItem, the story will have the title size 4 for stories with priority 1 through 3 and title size 2 for stories with priority 4 or lower.

```
[%{Imagelink}
<img src="<%Imagelink%>&MaxW=100" width=100 border=0 hspace=0 vspace=0 align=
"%][%{Counter is even&Imagelink}right">%][%{Counter is odd&Imagelink}left">%]
<font face=Verdana [%{TopStory} size=4> %|%  size=2> %]
<b><%\Title%></b></font><br>
<font face=Verdana size=1><%Leadin%></font><br>
<div align="left"><font face=Verdana size=1>
<a href="<%Link%>">Whole story</a> |
<a href="<%pbcs.dll%>/section?Category=<%Category%>">
<%Categoryname%></a></font></div>
<br>
</font>
```

Example news list

China Using wRights Issues To Promote Olympic Bid
Chinese leaders reacted indignantly seven years ago when Beijing's bid to host the Olympics was derailed by questions about human rights. They denounced critics for mixing sports and politics. One...
Whole story | World News
Md. Cemetery Gives Families Little Solace
Nicole Williams stares silently at the foot of the maple tree for a long while. This is where her fiance, Maurice Pinckney, was supposed to be buried after a senseless shooting cut short his life at...
Whole story | Regional News
Bill Passed To Sanction Lobbyists
The Maryland Senate voted for the first time yesterday to create a system of sanctions that could bar an unethical lobbyist from the halls of the State House for life.
Whole story | Regional News
Spies roam State posing as journalists, FBI says
Foreign intelligence agents working as news reporters retain sweeping unsupervised access to the State Department's central headquarters , despite a security crackdown after several major lapses ...
Whole story | National News
Georgetown Selects Non-Jesuit President
Georgetown University broke with tradition yesterday to select a president who is not a Jesuit priest, a first for the nation's oldest Roman Catholic university.
Whole story | Local News

Pbs macro

<pbs:macro>

Macros are functions related to object script. They allow you to build tags or output other code, based on URL parameters and options in the tag.

Template
This function can be used on all templates.

Folders
The system will look for the macro file in the following folders:
/<templates>/macros/<categoryid>
/<templates>/macros/<maincategoryid>
/<templates>/macros

Caching
You can run the tag uncached, so that it is cached independently of the template itself. When you write `<pbs:!macro.....>`, a separate cache file will be created. You must then use the cache time option to set the refresh interval.

Options

Before and After	The Before option makes sure that the tag is parsed before the story, or in whatever context you are using the macro. It is used like this: `<pbs:macro before name="macro">` Conversely, the After option makes sure that the tag is parsed after.
Cachequalifiers	This option is used for specifying which fields are needed to make the cache file unique so another page does not overwrite it with different content. The option is used like this: `<pbs:!macro cachetime=1440 CacheQualifiers="Date,Category,foo,site" name="macro">` This function can take fields from the URL and use them to create unique file names.
Cachetime	This sets the time the include should be cached before it is refreshed. This option is only used when the tag is running uncached.
Name	This option defines the file name for the macro. If you set the name to AP, the system will look for the macro file AP.pbo.

Noparse	This parameter should be used lightly. In the most extreme of circumstances, within a specific section of a website, it might be required to call a macro that then calls two or more other macros, which require the sharing of created global variables. Within this subsection of a website, where the two or more macros will need to share user-created variables, the noparse option should be added to the two or more second-level macro tags. This parameter ensures that the bottom level macros execute simultaneously and that any user-created variables on the same level are shared
Other options	You can define other variables by adding `field=value` to the tag. In addition, all options specified in the URL will be available in the macro file.When defining other variables, the names of the variables should not be words used elsewhere in SAXOTECH Online for field names, tag names, and so on.
Other variables	For fields are available in addition to system, URL, and global variables. If you put a macro on the result page, all fields in the form are available.

URL parameters	All parameters in the URL are available as fields. If you have the following URL:
	`/apps/pbcs.dll/article?Date=20020907&Category=NEWS02&ArtNo=221004&Ref=AR`
	You have these fields:
	`<%Date%>`
	with the value 20020907
	`<%Category%>`
	with the value NEWS02
	`<%Artno%>`
	with the value 221004
	`<%Ref%>`
	with the value AR

System variables
The following system variables are available, in addition to variables defined in the tag and taken from the URL. Please see System variables for more system variables.

<%CurrMainCategory%>	This field holds the current main category. It is written:`<%CurrMainCategory%>`

You can use your own variables in a format like the following:

```
<pbs:macro name="saxo" variable1="text1" variable2="text2">
```

Example 1
You want macro variables to contain the same values as the tags **<pbs:categoryname main>** and **<pbs:categoryname>.**

Use a macro tag like this:

```
<pbs:macro name=blah categoryname="<pbs:categoryname>">.
```

Then define a variable **<%categoryname%>** in the macro that is formatted with object script.

Example 2
You have the following tag on your front page:

<pbs:macro name=AP Lineup=HOME>

This will make the system look for the file AP.pbo, and the field Lineup in the object file. The lineup field has the value HOME.

The macro

```
<script language="JavaScript">
site="OHTOL";section="[%<%Section%>%|%<%Lineup%>%]";
</script>
<script language="JavaScript"
src="http://customwire.ap.org/lineups/<%Lineup%>.js">
</script>
```

The first script function sets the variable's site and section. These variables are going to be used by the included JavaScript. Section will be set to either the section or the lineup field, depending on what is used in the macro tag.The second line loads a JavaScript based on the lineup variable settings.

Example 3
The `<pbs:macro>` tag can be used for many things. For example, on election night when AP sends results in XML format, you can have a page with election results:

```
<pbs:!macro name="Election2008" Cachetime="15">
```

- Election2008.pbo

Election2008.pbo

ElectionScript contains some VBScript, which retrieves the election results from AP's FTP server.

In this way, election results from AP's elections service are updated every fifteen minutes.

```
<pbs:script name="ElectionScript" objectclass=1 Language="VBscript" file="xyz.xml">
```

Position-based design

Position-based design allows you to design the layout of each item in a newslist. To create object files for position based design, you add the position number to the end of the object file name. The object files for object class 1 must be named:

NewsItem1.pbo	Used for stories which have no position-based design.
NewsItem1_1.pbo	Used for position 1.
NewsItem1_2.pbo	Used for position 2.
NewsItem1_3.pbo	Used for position 3.

System variables

System variables are available in all object files. They are used to extract a query parameter from the URL for use in an object file or macro. This is done using a regular expression:

```
<%currentrequest$r(".*myParam\x3d(\w+).*=$1")V("myVariable")%>
```

The example above takes the value of the query parameter myParam and stores it in an objectscript variable called myVariable.

<%activedate%>	Contains the publishing date.
<%clientip%>	Contains the IP address of the computer that generated the page.
<%CurrCategory%>	Contains the current category ID. It must be run within a macro when used in article object files.

<%CurrMainCategory%>	Contains the current main category ID. It is written: `<%CurrMainCategory%>` It must be run within a macro when used in article object files.
<%CurrProfile%>	Contains the current profile ID. It must be run within a macro when used in article object files.
<%currentdate%>	Contains the current date.
<%CurrentDomain%>	Contains the domain from the Publicus.ini.
<%currentrequest%>	The request part of the URL. It is written: `<%currentrequest%>` Currentrequest will give you exactly what is in the request portion of the URL, including nocache=1 if it is present.
<%currentrequestmethod%>	Request method for the page. It is written: `<%currentrequestmethod%>`
<%currenturl%>	Contains the current URL.

<%inheritedlink%>	Contains the same value as
	`*<%currentrequest%>*`
	It is written:
	`<%inheritedlink%>`
	Inheritedlink will exclude the query parameter nocache=1.

<%ImageServer%>	Contains the value from the image server setting in Publicus.ini.

Chapter 7 Menu Tags

The menu tags are ways of building navigation using templates and objects rather than building each link manually. You can also put in tags manually where you want menu links. When you use the menu tags, you can turn entries on and off using settings in the category setup.

- pbs categorytree
  - CategoryTree.pbo
  - CategoryTreeItemActive.pbo
  - CategoryTreeItemPassive.pbo
  - CategoryTreeItemSeparatorColl.pbo
  - CategoryTreeItemSeparatorExp.pbo
  - CategoryTreeItemSeparatorNone.pbo
- pbs ft
- pbs leftmenu
- pbs menu-image
- pbs menu-text
- pbs profiletree
  - ProfileTree.pbo
  - ProfileTreeItem.pbo
  - ProfileTreeItemActive.pbo
  - ProfileTreeItemPassive.pbo
  - ProfileTreeItemSeparatorColl.pbo
  - ProfileTreeItemSeparatorExp.pbo
- pbs qwiknavsearch
  - QwikNavSearchForm.pbo

pbs categorytree

<pbs:categorytree>

This tag allows you to build a category tree showing all categories. This tag can be used on all templates.

Options

ExcludedCategoryList	Specifies the categories that the tag should not list. By default, this option is not used. To exclude a list of categories, enter:
	`ExcludedCategorylist=SPORTS01,SPORTS02,NEWS01,NEWS02,NEWS03`
	Wildcards can also be used:
	`ExcludedCategorylist=SPORTS%,NEWS%`

Expand	Controls which branches should be expanded. The expand option has the following values: • **All** - All branches are expanded. • **Active** - Only the active branch is expanded.
Levels	Specifies how many levels will be expanded by default. Usage: `Levels=3`
Object class	This option sets the object class for the object files.
Shownext	Expands all subcategories when a category is clicked.
Type	This option controls the type of category tree that is built. This option is set by using `type=xxx` The following values are available for xxx: • **upper** - The tree goes from the current category to the top of the tree. • **top** - The tree has all top level categories. • **complete** - A complete tree of all categories. • **lower** - The tree shows the category currently being viewed and its subcategories
Usesectioncat	Makes the function use the category in which the story is shown and not the category in which the story is placed. By default, the function uses the category in which the story is placed. Enter `usesectioncat=1` to use the category to which the profile is attached.

- CategoryTree.pbo
- CategoryTreeItemActive.pbo
- CategoryTreeItemPassive.pbo
- CategoryTreeItemSeparatorColl.pbo
- CategoryTreeItemSeparatorExp.pbo
- CategoryTreeItemSeparatorNone.pbo

CategoryTree.pbo

<%CategoryTreeContent%>	Contains the entire category tree.

CategoryTreeItemActive.pbo

Used for the category being viewed.

<%Name%>	Contains the name of the category.
<%OverView%>	Contains the link to the category.

CategoryTreeItemPassive.pbo

Used for the categories not currently being viewed.

<%Name%>	Contains the name of the category.
<%OverView%>	Contains the link to the category.

CategoryTreeItemSeparatorColl.pbo

This object is shown in front of all objects that contain a collapsed branch.

<%Expand%>	Contains the action necessary to expand the branch.

CategoryTreeItemSeparatorExp.pbo

This object is shown in front of all objects that contain an expanded branch.

<%Collapse%>	Contains the action needed to collapse the branch.

CategoryTreeItemSeparatorNone.pbo

This object is used in front of objects that cannot expand or collapse. This file has no fields.

pbs ft

<pbs:ft>

This tag was originally made for building the footer menu. The tag has morphed into the de facto tag for building menus. Menus can also be coded by hand using the anchor tag.

Options

The tag supports several options for modifying text style.

Activedot	Used to show which category is currently being viewed. When you use the activedot parameter, the menu must be enclosed in a table because the tag will output a table row for each item in the menu. Example:
	`<table border=0>`
	`<pbs:ft section=FRONTPAGE activedot="/images/arrow.gif" dotwidth=7 dotheight=11 nobold>`
	`<pbs:ft section=NEWS activedot="/images/arrow.gif" dotwidth=7 dotheight=11 nobold>`
	`</table>`
	The activedot parameter is the path to the image you would like to use.

DotHeight	Sets the height of the activedot.
DotWidth	Sets the width of the activedot.
Font Size	Controls the font size for the menu item. The tag will use the default font size. However, you can use font size from 1 to 6.

Front page	Creates a link to the front page. This option is not used by default. Example: `<pbs:ft frontpage>` When the option is used, the menu link goes to the front page. You can also link from the menu to the front page using the section option. To do this, you must first define the category Front page in the category register and set the link type to front page.

Lbgcolor	Sets the background color of the left cell. If the option is not set, the default background color will be used. This option can be set in two ways:
	• To use an image for the background, use the option
	`Lbcolor="img:/images/blue.gif"`
	• To use a color for the background, use
	`Lbcolor="#eeeeee"`

Nobold	Makes the link not bold. The links in the menu are bold by default.
Profile	Defines which profile should be used for the page. The option is set like this: `profile=1012`
Rbgcolor	Sets the background color of the right cell. This option has the same options as the Lbgcolor option. Notice that the option is called rbg and not rgb.
Section	Creates a link to a sectionfront. The parameter is the ID for the category. If the ID for the category is SPORT11 and the name is College basketball, you use `section=SPORT11` as the parameter.

Styleclass	This option is used for setting the style sheet class. The default class is Menu. Setting
	`styleclass=0`
	will turn off the use of styles. Setting
	`styleclass="Main"`
	will set the style sheet class to Main.

Target	Sets the target in a frame set.

pbs leftmenu

<pbs:leftmenu>

The Left menu tag is used for building graphical menus. This option is rarely used. The tag will use an image called **lm_categoryname.gif** in the directory **/grafikk**.

The tag supports the following options:

Forside	This option links to the front page.

Oversikt	This option links to a section front. Entering `oversikt=NEWS` links to the news section.

pbs menu-image

<pbs:menu-image>

This tag has the same functionality and options as the **<pbs:leftmenu>** tag. Please see the section for the pbs leftmenu tag for more information.

pbs menu-text

<pbs:menu-text>

This tag has the same functionality and options as the **<pbs:ft>** tag. Please see the section for the pbs ft tag for more information.

pbs profiletree

<pbs:profiletree>

This tag is used for building a tree of all profiles in the system. This tag can be used on all templates.

Options

Autoexpand	This will make the subcategories expand when a category is selected.
Category	Specifies the category like this: `category="news"` Please note that when you use <pbs:profiletree> as a navigation menu, the category must be specified. If you do not specify the category, the profile tree will be broken when a web guest goes to a category other than the one in the profile tree.
Excludeid	This option is used for excluding profile IDs that should not be part of the tree.Example: `excludeid="1000,1001,..."`
Object class	This option sets the object class for the object files.
Sectioncat	Sets the category in the link.
Topnode	This option is used for setting the top node for the profile tree. This enables you to build a profile tree for just one branch.

This function requires object files.

- ProfileTree.pbo
- ProfileTreeItem.pbo
- ProfileTreeItemActive.pbo
- ProfileTreeItemPassive.pbo
- ProfileTreeItemSeparatorColl.pbo
- ProfileTreeItemSeparatorExp.pbo

ProfileTree.pbo

If the ProfileTree.pbo does not exist, it will default to <%items%>, which will output the same result as earlier versions.

<%TopNode%>	The profile name of the ID. If no TopNode is given, it will be set to Root <%Items%>, which is the list of profile items.
<%TopNodeID%>	The TopNode ID given as input to the tag.

ProfileTreeItem.pbo

<%IsSubNode%>	This field has a value if the item is a sub node.
<%Item%>	Displays the item.
<%Separator%>	Displays the separator.

ProfileTreeItemActive.pbo

<%IsExpandable%>	This field has the value 1 if the item has sub items.
<%IsExtURL%>	This field has the value 1 if the item points to an extra URL.
<%ProfileName%>	Contains the name of the profile.
<%URL%>	Contains the URL for viewing the profile

ProfileTreeItemPassive.pbo

<%IsExpandable%>	This field has the value 1 if the item has sub items.
<%IsExtURL%>	This field has the value 1 if the item points to an extra URL.
<%ProfileName%>	Contains the name of the profile.
<%URL%>	Contains the URL for viewing the profile

ProfileTreeItemSeparatorColl.pbo

<%ExpandURL%>	This URL is used for expanding the branch.

ProfileTreeItemSeparatorExp.pbo

<%CollapseURL%>	This URL is used for collapsing the branch.

pbs qwiknavsearch

<pbs:qwiknavsearch>

This function can be used for jumping directly to a story or a sectionfront. This function is comparable to AOL keywords. This tag can be used on all templates.

To use this function, you must assign QwikNav words to the stories to which you would like to jump. You can also assign QwikNav words to aliases in the Alias database.

This tag has no options.

- QwikNavSearchForm.pbo

QwikNavSearchForm.pbo

<%Action%>	Contains the action for the form.
ErrorUrl	Used to set the error URL, which is used when the QwikNav search word does not exist.
QwikNav	Used to capture the QwikNav word.

The following is an example of the QwikNavSearchForm.pbo:

```
<form action="<%Action%>">
<b>QwikNav:</b><br>
<input type="hidden" name="Errorurl"
value="/apps/pbcs.dll/misc?Url=/templates/QwikNaverror.pbs">
<input type="text" name="QwikNav" size="9"
style="{font-family: Verdana, Geneva, Arial, Helvetica,
sans-serif; font-size: 11px; width: 80px;}"> 
<input type="submit" value="&raquo;"
style="{background-color: #efefef;
border: 1px solid Black;
font: 10px Verdana;}">
</form>
```

Chapter 8 Section Front Tags

The Sectionfront tags can only be used on the Sectionfront templates. These tags are used for building lists of stories. In addition to the Sectionfront tags, you can also use the overview tags to build lists of stories. The default Sectionfront template is called **gen_ovr.pbs**. For more information on template naming, please see Template names and URLs.

This tag is used for listing stories on sectionfronts. You should use the pbs newslist tag to build sectionfronts.

For the category setup set the overview type to 2, if the system should always show the sectionfront, even if there is only one story.

- pbs oversiktnotiser
- pbs sectionfront

pbs oversiktnotiser

<pbs:oversiktnotiser>

This tag is used for listing notices on the sectionfront of a category. Notices are used when you would like to avoid listing very short stories in the main overview. These stories must have the **Notice** checkbox marked in the story properties.

Options

Link	Controls the text in the story link. By default the tag will output [Read more]. Enter `link=????` where ???? is any text of your choice. The system will use this text instead for the link. Setting link=title makes the title the link.
Sortering	There are two possible sort options. By default, stories are sorted to show the newest stories first.Enter `sortoption=prio` to sort the stories according to priority.
Visref	Controls the number of summary characters that will be shown. By default the whole summary will be shown.Enter `leadin=#` to determine how many characters should be shown. This can be any number between 0 and 9999.

pbs sectionfront

<pbs:sectionfront>

This tag is used for listing stories on sectionfronts. You should use the <pbs:newslist> tag to build sectionfronts.

Output options

Count	Controls how many stories should be shown. By default, all stories published on today's date are shown. Enter `count=#` to determine a number of stories to be shown. This option overrides the archive setting in the category register.
Leadin	Controls the number of summary characters that will be shown. By default the whole summary will be shown. Enter `leadin=#` to determine how many characters should be shown. This can be any number between 0 and 9999.
Subcats	Controls whether or not stories from subcategories will be shown in the listing. By default, listing of stories from subcategories is turned off. Enter `subcats=1` to enable listing of stories from subcategories.
Summaries	Determines how many stories should have summaries. By default, all stories have summaries. Enter `summaries=x` to determine that only the first **x** stories will have summaries. Entering any number for **x** will show the summaries for the specified number of stories. Only the title will be shown for the other stories.

Image output options

Image	Controls if a thumbnail picture is shown in the story overview. This is a thumbnail of the main story picture. By default thumbnails are turned off. Entering `image=1` enables thumbnails.
Picturealign	Controls the placement of the thumbnail. By default, thumbnails are placed on the left. Enter `picturealign=right` to change the alignment of thumbnails to the right.
Picturepadding	Sets the padding on the thumbnail. By default, there is 4 pixels of padding around a thumbnail. Enter `picturepadding=#` to adjust the padding around the thumbnail. # can be any number between 0 and 100.
Picturewidth	Controls the width of the thumbnail. By default, all thumbnails are 80 pixels wide. Enter `picturewidth=#` to adjust the width of thumbnails. # can be any number between 10 and 1000.

Formatting options

Cathead	This option controls if the category/subcategory name should be shown above the title. By default this option is turned off. Enter `cathead=1` to enable this option.
Linebeak	Draws a line between each story in the list. By default, no line is drawn. Enter `linebreak=1` to draw a line between each story.
Link	Controls the text in the story link. By default the tag will output [Read more]. Enter `link=????` where ???? is any text of your choice. The system will use this text instead for the link. Setting link=title makes the title the link.

Linkcolor	Sets the link color. By default, the link color is the color set in the <body> tag. Enter
	`linkcolor=#????`
	to output a font color around the title in the following manner:
	`<font color=#????>`

Sortoption	There are two possible sort options. By default, stories are sorted to show the newest stories first. Enter `sortoption=prio` to sort the stories according to priority.

Subcatcolor	Sets the color of the subcategory header. By default, the text color is used. Enter
	`subcatcolor=#????`
	to output a font color around the subcategory name in the following manner:
	`<font color=#????>`

Titlefont	This option sets the font that will be used for the title. The default font face will be used.If you do not wish to use the default font face enter
	`titlefont=????`
	to output a different font face around the title in the following manner:
	`<font face=????>`

Titlesize	This option sets the font size that will be used for the title. By default the size is 4. This will output in the following manner:
	
	`<font size=4>`
	
	If you wish to change the font size, enter
	
	`titlesize=#`
	
	where # is a value between 1 and 6. This will output a different font size around the title in the following manner:
	
	`<font size=#>`

Togglecolumns	This option builds a page in which every other line contains two stories.

Chapter 9 Article Tags

Article tags are used to display stories.

- ArticleConnectionItem.pbo
- pbs articlefield
- pbs articledate
- pbs artikkel-urls
- pbs articlemultimedia
- pbs photogallery-links
- pbs keywordlinks
  - KeywordLinkItem.pbo
  - KeywordLinks.pbo
- pbs themelinks
  - ThemeLinkItem.pbo
  - ThemeLinks.pbo
- Other links
- pbs article-links
- pbs article
  - Article.pbo
  - ArticleByline.pbo
  - ArticleConnections.pbo
  - ArticleExtraPictureItem.pbo
  - ArticleExtraPictures.pbo
  - ArticleFacts.pbo
  - ArticleForumConnectionItem.pbo
  - ArticleForumConnections.pbo
  - ArticleGalleryConnectionItem.pbo
  - ArticleGalleryConnections.pbo
  - ArticleHead.pbo
  - ArticleLinkItem.pbo
  - ArticleLinks.pbo
  - ArticleLocation.pbo
  - ArticleModified.pbo
  - ArticleMultiMedia.pbo
  - ArticleMultiMediaExtension.pbo
  - ArticleMultimediaLinkItem.pbo
  - ArticleMultiMediaList.pbo
  - ArticleMultiMediaListItem.pbo
  - ArticleMultimediaListItemExtension.pbo
  - ArticleMultimediaListItemXXX.pbo
  - ArticleParagraph.pbo
  - ArticlePhotoGalleryThumbnailItem.pbo
  - ArticlePhotoGalleryThumbnails.pbo
  - ArticlePicture.pbo
  - ArticlePublished.pbo
  - ArticleSummary.pbo
  - InvalidArticle.pbo
  - Tips.pbo and articles
  - WAPArticle.pbo
  - WAPArticleContent.pbo
  - WAPArticleParagraph.pbo
- ArticleGeoDataItem.pbo
- ArticleGeoDataItems.pbo

ArticleConnectionItem.pbo

<%\Date%>	The publication date for the story.
<%Heading%>	The heading for the linked story.
<%Leadin%>	Outputs the summary if the leadin is larger than 0. To ensure that the leadin is larger than 0, use ShowRef.
<%Link%>	The address of the connected story.
<%TeaserUrl%> **From 6.3.0sr12**	The External URL of a teaser link, if available

pbs articlefield

<pbs:articlefield>

If your story interface is set up with extra fields like a quote or a slug, it is possible to extract these fields on the article template. Which fields you have available will vary depending on your site setup.

This tag can be used on all article templates.

Options

Field	The field option is used for extracting the field. It is used like this: `<pbs:articlefield field="xxx">` Where xxx is the name of the field. Extra fields can also be shown in the article object files using Field_XXX.

pbs articledate

<pbs:articledate>

This tag outputs the date the story was published.

Options

Dateformat	This option's parameters are described in Formatting dates.

pbs artikkel-urls

<pbs:artikkel-urls>

This tag outputs the same information as Links in the Article.pbo. You can also show links that have been tagged with the same keywords or theme using the links tag: `<pbs:links usetheme=1>`. The tag is documented here: pbs links.

Object files

The object file is described in the section ArticleLinks.pbo.

pbs articlemultimedia

<pbs:articlemultimedia>

This tag is used to output the code needed to show streamed media embedded in a pop-up window. This function is used when you turn off direct download on a linked asset.

This function uses the template **ArticleMultiMediaPopup.pbs**. The popup link in the object file ArticleMultiMediaListItem.pbo links to it.

This tag uses ArticleMultiMediaExtension.pbo, which is also used for the embedded streamed media.

This tag does not have any options.

pbs photogallery-links

<pbs:photogallery-links>

Used to show Gallery links. This tag outputs the same information as GalleryConncetionItems in the Article.pbo.

Object file

The object file is described in the section GalleryConncetionItems.

pbs keywordlinks

<pbs:keywordlinks>

Options

Antall	This option works the same way as **Count** on the Sectionfront tag. Please see Count
Visref	This option works the same way as **Leadin** on the Article tag. Please see Leadin.

- KeywordLinkItem.pbo
- KeywordLinks.pbo

KeywordLinkItem.pbo

The tag will use this object file if it exists. This file is used to build each item in the listing.

<%Counter%>	Countains the counter value.
<%\Date%>	The date the story was published.
<%Link%>	The URL that links to the story.
<%\Title%>	The title of the story.

KeywordLinks.pbo

The tag will use this object file if it exists.This file is used to build the framework around the stories.

<%Items%>	This is where the system will output the items.
<%MoreLink%>	This parameter will make links to all the stories with same keyword.

pbs themelinks

<pbs:themelinks>

This tag is used for theme links, also called related stories. It is used to display links to other stories that have been tagged with the same theme. You can also use ThemeLinks in the Article.pbo. The tag for displaying all related stories is described in the section Pbsrelated.

Options

Antall	This option works the same way as **Count** on the Sectionfront tag. Please see Count.
Bgcolor	Sets the background color for the header. This is only used when object files are not used. The default is the HTML color #CCCCCC. You can use any HTML color.

Color	Sets the text color in the header. This is only used when object files are not used. The default is the default color for the text page.This option is set by putting `color="#??????"` inside the tag.
Bredde	Sets the width to be used the by tag. Only used when object files are not used.
Fontsize	This option sets the font size. Only used when object files are not used. The default is `Fontsize=2.` You can use any size from 1 to 6.
Ledetekst	Sets the title for the listing. You can use any text. If you enter `ledektekst=""` it will not output any text.
Visref	This option works the same way as **Leadin** on the Article tag. Please see Leadin.

- ThemeLinkItem.pbo
- ThemeLinks.pbo

ThemeLinkItem.pbo

This file is used for building each item in the listing.

<%Counter%>	Countains the counter value.
<%\Date%>	The date the story was published. It can be formatted with object script.
<%Link%>	This is the URL that links to the story.
<%\Title%>	The title of the story.

ThemeLinks.pbo

This file is used for building the framework around the stories.

<%Items%>	This is where the system will output the items.
<%MoreLink%>	This parameter will make links to all the stories with same theme.

Other links

Linking to a keyword index

Use the following tag to link to the keyword index for a category:

```
<a href="<pbs:prog>/index?Category=<pbs:categoryid>">View the <pbs:category main> keyword
index</a>
```

pbs article-links

<pbs:article-links>

This tag is used to show stories that have been liked to the story being viewed. These stories are linked manually, through the web-based user interface. You can also use ConnectionItems in the Article.pbo.

Options

Forwardonly	This option will make the tag link only to the linked stories, and not from the linked story to the main story. Subarticles ignores this option.
Includegallery	This option will make the tag output links to photo galleries.

Object files

The tag uses ArticleConnections.pbo.

pbs article

<pbs:article>

This tag outputs the whole story or parts of it, depending on which options are used. Using the tag like this will output the story with default formatting. Some of the options on this tag block output of some elements of the story. The tag supports object files.

The article tag is used to display stories. Links to stories are automatically built by the overview tags. It is also possible to access a story by linking directly to it. The link will look something like this:

```
/pbcs.dll/artikkel?Avis=GE&Dato=20001008&Kategori=FRONTPAGE&Lopenr=11008068&Ref=AR
```

The article tag supports the use of article classes. This is a method of predefining different layouts for the article tag using object files. Please see Article classes.

The article tag also supports object files. Please see Object Files to learn more about object files.

The article tag supports the use of object files for controlling layout. You can also use different classes for the object files. Please see Article classes.

This tag is used on article templates. The general template is called **gen_art.pbs**.

Printer-friendly version

The print version of the story is a version that does not have any graphics. The print template is usually used in two different versions: One with pictures and the other without. The code used in the templates is stripped down and uses very little formatting.

The JavaScript in these templates will automatically open the print dialog box if the function is supported. If the function is not supported the text, *To print* *this article open the file menu and choose Print* is shown instead.

The plain text version uses the template **gen_printart.pbs**. The version with pictures uses the template `gen_printpicart.pbs.`. If you are using object files for the layout of stories, you must use the option UseObjects in the article tag on the print templates.

The <pbs:article> has the following options:

Object file options

Objectclass	The object class controls which object files the page will use. By default, the system will use the files in the default object class.Object class can be set from 1 to 999.
Useobject	This option turns off the use of **Article.pbo**. Object files are used for parts of the story instead. This is triggered by entering `useobject=0` in the article tag.

Useobjects	Object files are automatically used if they exist. To turn off the use of object files, add the option
	`useobjects=0`
	in the
	`<pbs:article>`
	tag.

Output options

Byline	Controls the output of the byline. The default is
	`Byline=1.`
	Enter
	`Byline=1`
	to enable output of the byline. Enter
	`Byline=0`
	to disable output of the byline.

Facts	If the story contains a fact box, setting
	`Facts=1`
	will output the fact box. The default is
	`Facts=0.`
	Enter
	`Facts=1`
	to enable output of the fact box. Enter
	`Facts=0`
	to disable output of the fact box. When using Facts as the only option, the tag will output only the fact box and no other parts of the story.

Header	Controls the output of the story header. The default is
	`Header=1.`
	Enter
	`Header=1`
	to enable output of the header. Enter
	`Header=0`
	to disable output of the header.

Karakter	Outputs the story's rating. The default is
	`Karakter=1.`
	Enter
	`Karakter=1`
	to enable output of the rating. Enter
	`Karakter=0`
	to disable output of the rating.

MediaOrder **available from 6.2.0**	This setting determines how the connected media (assets) are ordered `MediaOrder=(0|1|2)` The value specifies a sorting order: • 0 = Embeddable assets first, then direct linked assets first, then oldest assets first (default) • 1 = Embeddable assets first, then direct linked assets first, then newest assets first • 2 = Newest assets first, then embeddable assets first, then direct linked assets first

Modified	Outputs the story's last modification date. This option will use ArticleModified.pbo when you use the option
	`useobject=0.`
	The default is for the modification date to be set in the category register. Enter
	`Modified=1`
	to enable output of the modification date. Enter
	`Modified=0`
	to disable output of the modification date.

Published	Outputs the storyÕs publication date. This option will use ArticlePublished.pbo when you use the option
	`useobject=0.`
	The default is
	`Published=1.`
	Enter
	`Published=1`
	to enable output of the publication date. Enter
	`Published=0`
	to disable output of the publication date.

Summary	Controls the output of the story summary. The summary is shown in the summary field on the story. The default is
	`Summary=1.`
	Enter
	`Summary=1`
	to enable output of the summary. Enter
	`Summary=0`
	to disable output of the summary.

Tip	Controls the output of the basic Tip a Friend function. The function is controlled by the category setup, here you can override the category setting. The default is to enable output of the tip box in the category register.It is also possible to use the *article-tiplink *function to open a Tip a Friend window. Enter
	`Tip=1`
	to enable output of Tip a Friend. Requires that the option is enabled for the category.Enter
	`Tip=0`
	to disable output of Tip a Friend. When using Tip as the only option, only the tip box will be shown.

Text	Controls the output of the body text. The default is
	`Text=1.`
	Enter
	`Text=1`
	to enable output of the body text. Enter
	`Text=0`
	to disable output of the body text.

Title	Controls the output of the title. The default is
	`Title=1.`
	Enter
	`Title=1`
	to enable output of the title. Enter
	`Title=0`
	to disable output of the title.

Tittel2	Controls the output of the subtitle. The default is
	`Tittel2=1.`
	Enter
	`Tittel=1`
	to enable output of the subtitle. Enter
	`Tittel=0`
	to disable output of the subtitle.

Image options

Bilde	This option is the Norwegian version of the **Image** option.

Extrapictures	Controls output of the extra pictures imported on the story. You can also use a Photo Gallery. Please see Article linked photo gallery for information on Photo Galleries. The default is
	`Extrapictures=0.`
	Enter
	`Extrapictures=1`
	to enable output of extra pictures. Enter
	`Extrapictures=0`
	to disable output of extra pictures.

Image	Controls the output of paragraph photos. The default is
	`Image=1.`
	Enter
	`Image=1`
	to enables output of paragraph photos. Enter
	`Image=0`
	to disable output of paragraph photos. Only the main photo will be shown.

Paragraphpicturewidth	Sets the maximum width of paragraph pictures. This value is used by the image dll (**pbcsi.dll**) to resample high-resolution pictures uploaded to the system. The default is `Paragraphpicturewidth=0.` You can use any value between 1 and 999. If you set this value higher than the the width of the pictures you upload to the system, the system will not optimize the picture.

Picturealign	Controls the alignment of the main picture. The default is
	`Picturealign=Left.`
	Enter
	`Picturealign=Left`
	if the main pictures should be aligned on the left. Enter
	`Picturealign=Right`
	if the main pictures should be aligned on the right.

Picturebgcolor	Sets the background color for the picture and caption. The default is for no background color to be set.Any hex color triplet from #000000 to #FFFFFF. The value will be used as the background color for the table that holds the picture and the caption.

Pictureheight	Sets the maximum height of the main picture. This value is used by the image dll (**pbcsi.dll**) to resample high-resolution pictures uploaded to the system. The default is `Pictureheight=0.` You can use any value between 1 and 999. If you set this value higher than the the height of the pictures you upload to the system, the system will not optimize the picture.
Picturepadding	Controls how much padding the picture and caption have. This option is used by the cell padding parameter on the table. The default is `Picturepadding=1.` You can use any value between 0 to 99.
Picturewidth	Sets the maximum width of the main picture. This value is used by the image dll (**pbcsi.dll**) to resample high-resolution pictures uploaded to the system. The default is `Picturewidth=0.` You can use any value between 1 and 999. If you set this value higher than the the width of the pictures you upload to the system, the system will not optimize the picture.

Formatting options

These options control the layout of the story if object files are not used.

Faktawidth	Sets the width of the fact box in pixels. The default is `Faktawidth=120.` You can use any value between 10 and 1000.

Titlefont	Sets the font used for the story title. The string used here is used by the
	`<font face=ÓÓ>`
	tag output before the title. By default, the story inherits the document font.
	`Titlefont`
	can be set to any font or combinations of fonts.

Titlesize	This function sets the size of the story title. The legal values are 1-6. Setting the title size will result in the system outputting a `<font size=#>` where # is the value set.

Special options

adinsertspot	The ad insert function allows you to display an ad inserted into the story text. The function allows you to display any ad type in the story. The following code must be present in your article tag: `<pbs:article adinsertspot=SIDEAD02>` To show the ad, you must use AdInsert in your Article.pbo.

Gifoutput	Makes the system output the story as a GIF image instead of as text. This is used for making a preview for special access/pay sections of the site. This function creates a heavy system load and should be used sparingly. The default is
	`Gifoutput=0.`
	Entering
	`Gifoutput=1`
	enables output of the story in GIF format. Entering
	`Gifoutput=0`
	disables output of the story in GIF format.

Inherit	This option makes the article tag inherit the information about which story to show from the page that linked to it.
List	Makes the story output bullet points for each paragraph. The default is disabled. Enter `list="LIST"` in the tag to enable bullet points.
WAP	By entering `wap="WAP"` in the article tag, the following WAP object files will be used instead of the normal object files: • WAPArticle.pbo • WAPArticleContent.pbo • WAPArticleParagraph.pbo This will limit the length of the article page so it can be displayed by WAP devices. WAP stories behave like web stories. For example, images can be shown in both kinds of stories. However, large images do not work well with WAP.

- Article.pbo
- ArticleByline.pbo
- ArticleConnections.pbo
- ArticleExtraPictureItem.pbo
- ArticleExtraPictures.pbo
- ArticleFacts.pbo
- ArticleForumConnectionItem.pbo
- ArticleForumConnections.pbo
- ArticleGalleryConnectionItem.pbo
- ArticleGalleryConnections.pbo
- ArticleHead.pbo
- ArticleLinkItem.pbo
- ArticleLinks.pbo
- ArticleLocation.pbo
- ArticleModified.pbo
- ArticleMultiMedia.pbo
- ArticleMultiMediaExtension.pbo
- ArticleMultimediaLinkItem.pbo
- ArticleMultiMediaList.pbo
- ArticleMultiMediaListItem.pbo
- ArticleMultimediaListItemExtension.pbo

- ArticleMultimediaListItemXXX.pbo
- ArticleParagraph.pbo
- ArticlePhotoGalleryThumbnailItem.pbo
- ArticlePhotoGalleryThumbnails.pbo
- ArticlePicture.pbo
- ArticlePublished.pbo
- ArticleSummary.pbo
- InvalidArticle.pbo
- Tips.pbo and articles
- WAPArticle.pbo
- WAPArticleContent.pbo
- WAPArticleParagraph.pbo

Article.pbo

This is the main object file for the story. It supports the following variables:

<%AccessControlIdentifier%>	Contains the flow ID that is associated with a story in the edit interface. If there are no flows associated with the story, then this variable will be empty. This variable is used in conjunction with Access Control.
AdInsert	The ad insert function allows you to display an ad inserted into the story text. The function allows you to display any ad type in the story. This function requires a special setting in the Publicus.ini. Please see Adinsert for more information.
<%ArtNo%>	Contains the story number.
<%ArticleStatus%>	Used to show different text based on the story status. This example shows how you can use the article status field to show different messages: `[%{ArticleStatus=0}` `  This story is not yet online.` `%]` `[%{ArticleStatus=1}` `  This story has been unpublished and is no longer available.` `%]` `[%{ArticleStatus=7}` `  This story is not available, because we do not have the rights to publish it online.` `%]` `[%{ArticleStatus=8}` `  This story has been archived and is only available from our paid archive.` `   ` `  <a href= "<%inheritedlink%>&Subscribertype=ARCHIVED&Sectioncat=PROTECTED" >Click here to log in to the archive</a> ` `%]` `[%{ArticleStatus=9|!Articlestatus}` `  The story goes here.....` `%]`
<%AssociatedKeywordLinks%>	Outputs external links that have the same keywords as the story. The following options are supported in the field: Count, Main category, Subcategory and Object class. This field has an attribute to specify which subcategory you would like to show. Example: `<%AssociatedKeywordLinks MainCategory=177987 SubCategory=10005 Count=5 ObjectClass=30%>`
<%AssociatedThemeLinks%>	Outputs external links that have the same theme as the story. The following options are supported in the field: Count, Main category and Object class.
<%AvgRanking%> **From version 6.4.0**	Contains the average ranking of a story.
<%Byline%>	Outputs the byline. It uses the file ArticleByline.pbo.
<%Category%>	Contains the category ID.

<%ConnectedEmbedMedia%>	Used to embed multimedia in the story. This field uses ArticleMultiMediaExtension.pbo.
	
	`[Extension]`
	
	is replaced by the media type. It is written:
	
	`<%ConnectedEmbedMedia%>`

<%ConnectedMedia%>	Lists connected multimedia objects. This field uses the object file ArticleMultiMediaList.pbo. It is written:
	
	`<%ConnectedMedia%>`
	
	For example,
	
	`<%ConnectedMedia%>`
	
	in **article50.pbo** searches for pbo files in the following order:
	
	`articleMultiMediaListItemPDF50.pbo`
	`articleMultiMediaListItemPDF.pbo`
	`articleMultiMediaListItem50.pbo`
	`articleMultiMediaListItem.pbo`

<%ConnectionItems%>	Outputs the links for connected stories, forums and galleries. It uses the files ArticleConnections.pbo and ArticleConnectionItem.pbo. Add following, to the <%ConnectionItems ShowRef=250%> to ensure that the leadin variable will contain summary text.
<%CurrentFlowStatusID%>	Contains the current status of the story within a flow using an ID. This variable is used in conjunction with Access Control.
<%CurrentFlowStatusText%>	Contains the current status of the story within a flow using text. This variable is used in conjunction with Access Control.

<%DesignClass%>	The story's class.
	
	`[%{DesignClass=a}<h2>Breaking News</h2>%]`
	`[%{DesignClass=b}<b>Normal article</b>%]`

<%ExtraFieldType%>	Contains the chosen type for the article. Types are defined in the variable EXTRAFIELDDEF section in the Publicus.ini.
<%ExtraPictures%>	Extracts extra pictures attached to a story. These pictures must be imported from your editorial system. To use this variable you must have the object files ArticleExtraPictures.pbo and ArticleExtraPictureItem.pbo.
<%Facts%>	Outputs the main fact box for the story. It uses the file ArticleFacts.pbo.
<%Field_xxx%>	Extracts extra fields from the story. Replace `xxx` with the name of the field.
<%ForumConnectionItems%>	Outputs the links for connected forums. It uses ArticleForumConnections.pbo and ArticleForumConnectionItem.pbo. It is written: `<%ForumConnectionItems%>`

<%GalleryConnectionItems%>	Outputs the links to the connected galleries. It uses the files ArticleGalleryConnections.pbo and ArticleGalleryConnectionItem.pbo.
<%Head%>	Outputs the story head. It uses the ArticleHead.pbo file.
<%Inheritedlink%>	Contains the link to the page. It can be used for building a link to a printer friendly page like this: `<a href="<%inheritedlink%>&template=printart" >Printer-friendly</a>`
<%IsoModifiedDate%>	The date the story was last modified in ISO format. It is written: `<%IsoModifiedDate%>`
<%ISOPublishedDate%>	The publication date for the story in ISO format. It is written: `<%ISOPublishedDate%>` If publish control are activated and this publish control have a newer date then published date, will the value of publish control take over.

<%KeywordLinks%>	Outputs links to other stories that have the same keywords as the story. You set the count option like this:
	`<%KeyWordLinks count=10%>`
	To control the number of months the tag looks back, you use the count option like this:
	`<%KeywordLinks months=5%>`
	For more information please see pbs keywordlinks.

<%Keywords%>	Contains the keywords attached to the story.
<%Links%>	Outputs the external links. It uses the files ArticleLinks.pbo and ArticleLinkItem.pbo. You can also show links that have the same keywords or theme as the story.
<%MainPicture%>	Outputs the main story picture. It uses the file ArticlePicture.pbo.
<%ModifiedDate%>	Outputs the latest modification date for the story. This field does not use ArticleModified.pbo. You can use the field like this: `Published: <%PublishedDate%>[%{ISOModifiedDate > #ISOPublishedDate#} - Last modified: <%ModifiedDate%>%] .`
<%NewsGroupAutolink%>	Outputs the link to the forum, if the story has forum. This option uses the Autoforum parameter. It is written: `<%NewsGroupAutolink%>`
<%ParagraphItems%>	Outputs all of the paragraphs for the story. It uses the file ArticleParagraph.pbo.
<%PublishedDate%>	The publication date for the story. The `<%PublishedDate%>` field does not use ArticlePublished.pbo.

<%RankingLink%> **From version 6.4.0**	The RankingLink does not contain the actual ranking value this must be added to the link by using client side scripting. The Following must be added to the link before submitting the link to the DLL **<%RankingLink%>** + '&rating=' + y Where y is the ranking given by a user.
<%Rating%>	Outputs the rating.
<%SingleSaleID%>	Contains the single sale ID associated to a story. It is possible to associate a single sale id to a story in the editing interface. This is used for Access Control.
<%Site%>	Contains the site code.
<%Summary%>	Outputs the summary for the story. It uses the file ArticleSummary.pbo.
<%Theme%>	Contains the story theme.

<%ThemeLinks%>	Outputs the links to other stories that have the same theme. To control the number of stories listed, use the count option like this:
	`<%ThemeLinks count=5%>`
	To control the number of months the tag looks back, use the count option like this:
	`<%ThemeLinks months=5%>`
	Please see pbs themelinks for more information.

<%ThumbnailGallery%>	Displays the first photo gallery attached to the story as thumbnails. This tag supports the option for setting columns, rows and object class like this: `<%ThumbnailGallery columns=4 rows=1 objectclass=1%>` The field uses the files ArticlePhotoGalleryThumbnails.pbo and ArticlePhotoGalleryThumbnailItems.pbo. These files have the same variables as the non-story versions documented in pbs photogallery-thumbnails
<%TipLink%>	Contains the link to the tip form. For more information please see Tip functions.
<%Tips%>	Outputs the integrated tip form. For more information please see Tip functions.
<%TotalNoofVotes%> **From version 6.4.0**	Contains the total number of votes on the story.
<%GeoDataItems%>	Output the geo data (locations on a map) attached to the story. For more information please see ArticleGeoDataItems.pbo and ArticleGeoDataItem.pbo.

ArticleByline.pbo

The byline file operates in different modes depending on category settings, whether or not the person is found in the staff register, and whether or not Smart Bylines are enabled. To use the Smart Bylines feature, you must use `ArticleByline.pbo`.

If the Smart Bylines feature is enabled, the byline will show the first name, surname, and e-mail address of the staff member (as entered in the Staff component) selected in the user interface.

If Smart Bylines is not used, a byline is only processed if it is active and in the category register. If the byline is set to active, you can use fields from the staff register to format the byline. This can be overridden by entering

```
AlwaysUseBylineObject=1
```

in the Publicus.ini. Alternatively, you can make the system output the original byline written in the story.

When not using smart bylines, the Online Robot will try to find staff information based on first name and last name in the byline field. This requires removing noise words from the byline field before processing. Noise words that must be removed from the byline field are defined in OnlineRobot_StoryExport.ini.

```
[[Sitecode_SITE_]SETTINGS]
NoiseWordsInByline="BY "," and ","Staff Writer", "#13#10"
```

Online starts from left to right and combines 2 words (firstname + ' ' + lastname) at a time before looking up the staff.

<%Adress1%>	The first part of the address as defined in the staff registry.
<%Adress2%>	The second part of the address as defined in the staff registry.
<%Adress3%>	The third part of the address as defined in the staff registry.
<%Adress4%>	The fourth part of the address as defined in the staff registry.
<%Byline%>	Outputs the names of all authors with the names as links to the e-mail addresses of those authors the system finds in the register. The function also outputs the other lines in the story.
<%BylineEmailPart%>	Contains the author's e-mail address as saved in the staff register. The same functions as for **<%BylineNamePart%>** are available here.
<%BylineNamePart%>	Contains the author's name as registered in the staff register. If you have two names in the byline you also have the field **<%BylineNamePart2%>**. This function supports up to 9 names in the byline.
<%City%>	Used for Smart Bylines. Contains the author's city as saved in the staff register.
<%Department%>	Used for Smart Bylines. Contains the author's department as saved in the staff register.
<%DesignClass%>	Contains the design class used.
<%Extension%>	Used for Smart Bylines. Contains the author's work phone number extension as saved in the staff register.
<%Fax%>	Used for Smart Bylines. Contains the author's fax number as saved in the staff register.
<%Firstname%>	Used for Smart Bylines. Contains the author's first name as saved in the staff register.
<%Fylkenr%>	Used for Smart Bylines. Contains the author's fylkenr as saved in the staff register.
<%Group%>	Used for Smart Bylines. Contains the author's group as saved in the staff register.
<%ImgLink%>	Contains the URL of the author's picture. For additional authors use the field **<%ImgLink2%>**. This tag supports up to 9 people.
<%Information%>	Used for Smart Bylines. Contains the author's information as saved in the staff register.
<%IsPrimary%>	Used for Smart Bylines. `IsPrimary=1` indicates that the author is the primary author.
<%Kommunenr%>	Used for Smart Bylines. Contains the author's kommunenr as saved in the staff register.
<%LastName%>	Used for smart bylines. Contains the author's surname as saved in the staff register.
<%Line1%>	Contains the first line of the byline field.
<%Line2%> etc...	You can increase the line number to output the different lines of the byline field.
<%MobilePhone%>	Used for Smart Bylines. Contains the author's mobile phone number as saved in the staff register.
<%Name%>	Outputs the names of all authors without e-mail links. The function also outputs the other lines in the story.
<%Namecount%>	Contains the number of authors the system has recognized. It can be used like this: `[%{namecount=1}<%BylineNamePart1%>%|%<%BylineNamePart%> and <%BylineNamePart2%>%]`
<%Office%>	Used for Smart Bylines. Contains the author's office as saved in the staff register.
<%PagerNumber%>	Used for Smart Bylines. Contains the author's pager number as saved in the staff register.

<%Position%>	Used for Smart Bylines. Contains the author's position as saved in the staff register.
<%PrivateEmailAdress%>	Used for Smart Bylines. Contains the author's private email address as saved in the staff register.
<%PrivatePhone%>	Used for Smart Bylines. Contains the author's private phone number as saved in the staff register.
<%PrivateWebAdress%>	Used for Smart Bylines. Contains the author's private web address as saved in the staff register.

<%StaffID%>	Contains the value from the staff ID field. This allows you to link to the information page for one author. The field is used like this:
	`<a href="/apps/pbcs.dll/personalia?ID=<%StaffID%>">`
	`<%BylineName Part%></a>`
	Please see pbs staffinfo for more information.

<%Surname%>	Used for Smart Bylines. Contains the author's surname as saved in the staff register.
<%WorkEmailbAdress%>	Used for Smart Bylines. Contains the author's work email address as saved in the staff register.
<%WorkPhone%>	Used for Smart Bylines. Contains the author's work phone number as saved in the staff register.
<%WorkWebAdress%>	Used for Smart Bylines. Contains the author's work web address as saved in the staff register.
<%Zipcode%>	Used for Smart Bylines. Contains the author's zip code as saved in the staff register.

ArticleConnections.pbo

<%ConnectionItems%>	Outputs the story items.
<%ForumConnectionItems%>	Outputs the forum items.
<%GalleryConnectionItems%>	Outputs the photo gallery items.

ArticleExtraPictureItem.pbo

<%ArtNo%>	Contains the image ID number.
<%Category%>	The image category.
<%Counter%>	Contains the picture counter. It starts on 1 for the first picture.
<%DesignClass%>	Contains the design class used.
<%Img%>	HTML image tag to make sure the image is displayed.
<%IsLastItem%>	This field has the value 1 on the last picture.
<%ISODate%>	The date of the image.
<%Site%>	The two letter site code.

All fields from ArticlePicture.pbo are also supported.

ArticleExtraPictures.pbo

<%Count%>	Shows the number of pictures.
<%DesignClass%>	Contains the design class used.
<%Items%>	Outputs the items.

ArticleFacts.pbo

Used to define the fact box. It is used for all fact boxes in the story. The file `ArticleFacts_0.pbo` will only be used for the main fact box. `ArticleFacts_1.pbo` will be used for the fact box in the first paragraph, `ArticleFacts_2.pbo` for the second paragraph, and so on. `ArticleFacts.pbo` is used for all fact boxes that do not have their own template.

<%Align%>	Outputs the alignment of the fact box. In the editing interface, users can select either left or right.
<%DesignClass%>	Contains the design class used.
<%Heading%>	Outputs the title for the fact box.
<%Text%>	Outputs the text for the fact box.
<%Width%>	Outputs the value of the FaktaWidth option in the article tag, which is used for controlling the width of the fact box.

ArticleForumConnectionItem.pbo

This file is used by **<%ForumConnectionItems%>** in both Article.pbo and ArticleConnections.pbo.

<%\Date%>	The creation date for the forum.
<%Link%>	The address of the connected forum.
<%ObjectClass%>	If you need to use the current object class with the forum template, add it to the link like this: `<%Link%>&ObjectClass=<%ObjectClass%>`
<%\Title%>	The heading of the linked forum.

ArticleForumConnections.pbo

This file is used with **<%ForumConnectionItems%>** in Article.pbo.

<%ConnectionItems%>	Outputs the forum items.

ArticleGalleryConnectionItem.pbo

<%ImgLink%> **From version 6.2.0**	Contains the link to the first image in the gallery.
<%Link%>	Contains the link to the gallery.
<%\Title%>	Contains the title of the gallery.

ArticleGalleryConnections.pbo

Used when a gallery is attached to a story.

<%ConnectionItems%>	Shows all gallery all items.

ArticleHead.pbo

Controls output of the story header. It supports the following fields.

<%ArtNo%>	Contains the story number.
<%Category%>	Contains the category ID.
<%DesignClass%>	Contains the design class used.
<%Heading%>	Outputs the main heading.
<%Heading2%>	Outputs the subheading.
<%Site%>	Contains the site code.

<%SuperHeading%>	Outputs the super heading (Kicker Field). You can use object scripts to force capitalization of the super heading. For example:
	`<%Superheading$u%>`
	will show as 'THIS IS A TEST'
	`<%Superheading$p%>`
	will show as 'This is a test'
	`<%Superheading$n%>`
	will show as 'This Is A Test'
	`<%Superheading$l%>`
	will show as 'this is a test'

ArticleLinkItem.pbo

This file is output for each of the external links in the story.

<%AgeRating%> **From version 6.2.0**	Contains the age rating.
<%Counter%>	The counter is the number for the link. The fist link is given the value 1 and the second link the value 2. This value is increased by 1 for each additional link.
<%Description%>	Contains the description of the site.
<%DesignClass%>	Contains the design class used.
<%ImageLink%>	Contains a link to the photo defined in Story External Links in the Edit interface.
<%Link%>	Contains the link to the external site.
<%LinkID%<	Contains the link's ID.
<%LinkMainCategory%>	Contains the main category of the linked story.
<%LinkMainCategoryID%>	Contains the main category ID of the linked story.
<%LinkRegion%>	Contains the region of the linked story.
<%LinkSubCategory%>	Contains the subcategory of the linked story.
<%LinkSubCategoryID%>	Contains the subcategory ID of the linked story.
<%LinkSubRegion%>	Contains the subregion of the linked story.
<%LogoLink%>	Contains a link to the logo defined in External Links in the edit interface.
<%Name%>	Contains the name of the external site.
<%QualityRating%> **From version 6.2.0**	Contains the quality rating.

ArticleLinks.pbo

Used when you have links to external sites in the story.

<%DesignClass%>	Contains the design class used.
<%LinkItems%>	This field outputs all the links.

ArticleLocation.pbo

Controls output of the dateline.

<%DesignClass%>	Contains the design class used.
<%Location%>	This field contains the dateline.

ArticleModified.pbo

This object file is only used when you use Useobject.

<%DesignClass%>	Contains the design class used.
<%ISOModifiedDate%>	Contains the publication date in ISO format.
<%ModifiedDate%>	The publication date.

ArticleMultiMedia.pbo

The fallback file used if you have not created an object file for the media type. Because different file types are managed differently, it is nearly impossible to write a general object file for all media types. This file should tell the user that the support file for the media format is not installed on the server.

<%DesignClass%>	Contains the design class used.

ArticleMultiMediaExtension.pbo

ArticleMultiMedia[Extension].pbo

Used for embedded multimedia objects.

```
[Extension]
```

is replaced by the media type extension. See the example section for the structure for different formats. This file is also used by the tag **<pbs:articlemultimedia>**

<%Align%>	Contains the align link from the story properties page.
<%AutoStart%>	Contains 0 or 1, depending on whether or not the file should auto start.
<%DirectDownload%>	Has the value 1 if the item is to be downloaded directly.
<%DesignClass%>	Contains the design class used.
<%DirectLink%>	The direct link to the file.
<%ExternalURL%>	Contains the external URL of the item.
<%FileDescription%>	A description of the object.
<%FileName%>	The filename of the object.
<%FileSize%>	The size of the object.
<%Height%>	Contains the height from the properties page for the media type.
<%Link%>	Contains the link to the item.
<%MediaDescription%>	The description of the media type.
<%OrgFileName%>	The original filename.
<%Start%>	The start field.
<%Stopp%>	The stop field.
<%StreamLink%>	Contains the stream link to the item.
<%Width%>	Contains the width from the properties page for the media type.

ArticleMultimediaLinkItem.pbo

Used for linking to movie files.

<%Description%>	Contains a description of the content.
<%DesignClass%>	Contains the design class used.
<%Format%>	Contains the name of the format.
<%MovieLink%>	Contains the link to the movie file.
<%Speed%>	Contains the speed value necessary to view the movie stream.

ArticleMultiMediaList.pbo

Lists the connected media items.

<%MediaItems%>	Lists all the connected items.

ArticleMultiMediaListItem.pbo

Used for all connected items.

<%AutoStart%>	Contains either 1 or 0. Controls whether or not embedded clips auto start.
<%Counter%>	Countains the counter value.
<%DirectDownload%>	Has the value 1 if the item is to be downloaded directly.
<%DirectLink%>	Contains the link for direct download.
<%Embed%>	Has a value if the file can be embedded. This is defined in media type setup.
<%FileDescription%>	A description of the file.
<%FileName%>	The internal file name.
<%FileSize%>	The file size in bytes.
<%Link%>	The link to the item.
<%MediaDescription%>	A description of the media type.
<%OrgFileName%>	The original file name.
<%PopupLink%>	Contains the link to the popup page. This page is used if the item is not to be downloaded directly.
<%Start%>	Controls the start point in the clip.
<%Stopp%>	Controls the end point in the clip.
<%StreamLink%>	The streaming link to the item.

ArticleMultimediaListItemExtension.pbo

ArticleMultimediaListItem[Extension].pbo

You can override the object file used for each media type. This is an optional feature for special cases.

ArticleMultimediaListItemXXX.pbo

Where XXX is your Castfire video asset type. Used for Castfire integration.

```
<embed
id="castfire_player_<%OrgFileName%>_flv"
class="castfire_player"
src="<%Link%>"
quality="high"
wmode="transparent"
name="castfire_player_<%OrgFileName%>_flv"
allowScriptAccess="always"
type="application/x-shockwave-flash"
pluginspage="http://www.macromedia.com/go/getflashplayer"
width="<%Width%>"
height="<%Height%>">
</embed>
```

ArticleParagraph.pbo

<%AltText%>	Outputs the information from the image Alt text field.
<%Blockid%>	Contains the ID for the paragraph. The first paragraph is given the ID 1. The number is increased by one for each paragraph.
<%Body%>	Outputs the paragraph.
<%DesignClass%>	Contains the design class used.
<%Facts%>	Outputs the fact box for the paragraph.
<%Heading%>	Outputs the heading for the paragraph.
<%Html%>	This field has the value 1 if the HTML flag is turned on for the paragraph. Otherwise, the field is blank.
<%Location%>	Outputs the dateline. To limit it to the first paragraph, you can use the following code: `[%{location&BlockID=1}<%location%>%]` Please see Stedopt.
<%Picture%>	Outputs the paragraph picture.

ArticlePhotoGalleryThumbnailItem.pbo

This file has the same variables as the non-story version documented in the chapter pbs photogallery-thumbnails.

ArticlePhotoGalleryThumbnails.pbo

This file has the same variables as the non-story version documented in the chapter pbs photogallery-thumbnails.

ArticlePicture.pbo

Used to display the pictures on the story. You can also use numbering in the file name to determine which pictures will use the file. The file ArticlePicture_0.pbo will only be used for the main picture. ArticlePicture_1.pbo will be used for the picture on the first paragraph, ArticlePicture_2.pbo for the second paragraph, and so on. ArticlePicture.pbo is used for all pictures that do not have their own template.

If you have integration with QPS, you might get pictures that contain only the caption and not the picture itself. These images have the size -1 by -1 pixels. You can use object script to check that the size is larger than 1 to avoid outputting just the caption.

<%Align%>	This field contains the value from the editing interface for aligning the picture left or right. For the main picture, it uses the value from the PictureAlign option in the article tag. It is usually used like this in the table that wraps the picture: `<table align="<%Align%>">`

<%AltText%>	Outputs the information from the image Alt text field.
<%Caption%>	The caption for the picture.
<%CaptionTitle%>	The title for the picture.
<%DesignClass%>	Contains the design class used.
<%Height%>	Works like **<%width%>** for height.

<%ImgKeyParams%>	Contains the key parameters for the picture. These parameters are used when you want to use functions like zoom. It is usually used like this:
	`<a href="/apps/pbcs.dll/misc?url=/templates/zoom.pbs&`
	`<%ImgKeyParams%>">Zoom</a>`
	This function is described in the section pbs link and images.

<%ImgLink%>	Contains the URL that is used as source for the **<img>** tag. It is possible to use the options **Title=1** and **Logo=/somefile.gif** in conjunction with this field so that the tag can be used like this:
	`<img src="<%ImgLink%><%MaxWidth%>&`
	`title=1&logo=/graphics/logo.gif">`

<%MainPicture%>	This field has the value 1 if the picture is the main picture.

<%MaxHeight%>	Contains the value from the options **PictureHeight** or **ParagraphPictureHeight**. It outputs
	`&MaxH=XXX`
	where
	`XXX`
	is the value from the option. It is used with the **<%ImgLink%>** field.

<%MaxWidth%>	Contains the value from the options PictureWidth or ParagraphPictureWidth. It outputs
	`&MaxW=XXX`
	where
	`XXX`
	is the value from the option. It is used with the **<%ImgLink%>** field.

<%Orientation%>	This field contains the value
	`L`
	if the image orientation is landscape and
	`P`
	if it is portrait.

<%Photo%>	The photo byline.

<%Width%>	Contains the width of the picture. **<%width%>** outputs the value passed in from
	
	`<pbs:article picturewidth=XXX>`
	
	if
	
	`XXX`
	
	is defined and is less than the image's actual width. If the image's width is larger, it outputs the actual width.It can be used with the **<%ImgLink%>** field or in the table tag like this:
	
	`<table width="<%Width#+8%>">`
	
	When you use the option like this, you will get spacing around the picture table.

ArticlePublished.pbo

This object file is only used when you use Useobject.

<%DesignClass%>	Contains the design class used.
<%ISOPublishedDate%>	Contains the publication date in ISO format.
<%PublishedDate%>	Contains the publication date.

ArticleSummary.pbo

Controls output of the story summary.

<%DesignClass%>	Contains the design class used.
<%Location%>	Outputs the content of the location field.
<%\Summary%>	Outputs the content of the summary field.

InvalidArticle.pbo

If a web guest links to a story that no longer exists, she will get see error message: *Invalid article key.* You can override this so that the web guest will see a more understandable error message.

<%Message%>	The message from the system.
<%StoryCategory%>	The category in which the story is placed.
<%StoryDate%>	The story's date.
<%StoryID%>	The story's ID.
<%StorySite%>	The site on which the story is published.

Tips.pbo and articles

For information about the integrated tip form, see the section pbs article-tiplink.

WAPArticle.pbo

This object file is triggered when the pbs article#WAP option is used on the article tag.

<%Category%>	Contains the category name.
<%Content%>	Contains the content.

<%NextPageLink%>	Contains the link to the next page. Because a WAP phone can only display a small text block, the story is split up into multiple parts.
<%SectionLink%>	Contains a link back to the section front.

WAPArticleContent.pbo

This object file is triggered when the pbs article#WAP option is used on the article tag.

<%Byline%>	Contains the byline for the story.
<%DesignClass%>	Contains the design class used.
<%Heading%>	Contains the heading for the story.
<%ModifiedDate%>	Contains the date the story was last modified.
<%Paragraphs%>	Contains all the text blocks.
<%PublishedDate%>	Contains the publication date. It is written: `<%PublishedDate%>`
<%Source%>	Contains the source name for the story.
<%\Summary%>	Contains the summary.

WAPArticleParagraph.pbo

This object file is triggered when the pbs article#WAP option is used on the article tag.

<%Body%>	Contains the body text of the text block.
<%DesignClass%>	Contains the design class used.
<%Heading%>	Contains the heading for the text block.

ArticleGeoDataItem.pbo

Geo Data is available in SAXOTECH Online from version 6.3.0.

<%Name%>	Contains the name of the geo data item, as for example "The summer house"
<%Location%>	The address or other text representation of the location, as for example "Petersbuger Str. 91, Berlin"
<%Latitude%>	The latitude of the location, if known. Latitude will be a number between -90 and 90. If unknown it is blank (not to be confused with 0 which is a valid, although rarely used, latitude.)
<%Longitude%>	The longitude of the location, if known. Longitude will be a number between -180 and 180. If unknown it is blank (not to be confused with 0 which is a valid, although rarely used, longitude.)

ArticleGeoDataItems.pbo

Geo Data is available in SAXOTECH Online from version 6.3.0.

<%ArticleGeoDataItems%>	Outputs all Geo data items attached to the story.

The individual Geo Data items are rendered using ArticleGeoDataItem.pbo.

Chapter 10 Overview Tags

The tags in this chapter are used for building overviews of stories. This is similar to a section front and can be used as such. They can be used on most templates.

- pbs agent
  - Agent.pbo
  - AgentItem.pbo
- pbs agentcategory
- pbs history
  - History.pbo
  - HistoryItem.pbo
- pbs keywordindex
  - KeywordIndexHeader.pbo
  - KeywordIndexLink.pbo
  - KeywordIndexTitle.pbo
- pbs mostread
  - MostRead.pbo
  - MostReadItem.pbo
- pbs newslist
  - News.pbo
  - NewsItem.pbo
  - NewsItemArticleConnectionItem.pbo
  - NewsItemConnectionItem.pbo
  - NewsItemExternalLinkItem.pbo
  - NewsItemExternalLinks.pbo
  - NewsItemForumConnectionItem.pbo
  - NewsItemForumGroupLink.pbo
  - NewsItemGalleryConnectionItem.pbo
  - NewsItemInternalLinks.pbo
  - NewsItemKeywordLinks.pbo
  - NewsItemKeywordLinksItem.pbo
  - NewsItemMultiMediaExtension.pbo
  - NewsItemThemeLinkItem.pbo
  - NewsItemThemeLinks.pbo
  - RelStoriesItem.pbo
  - RelStoriesItems.pbo
  - URL Parameters and overview tags
- pbs related
  - ThemeLinkItem.pbo and overview tags
  - ThemeLinks.pbo and pbs related
- pbs subcategories
  - SubcatSimpleLink.pbo

pbs agent

<pbs:agent>

This tag is used for listing stories that come into the system from press agencies. The tag was previously used to list the most popular stories on a site, functionality that has been moved to the pbs mostread tag.

This is an example of what the tag could look like:

```
<pbs:agent lead-in=0 count=5 category="ntbu">
```

This tag will use object files if they exist. Object files are **not** available when you use the **Channels** option in the Publicus.ini.

ArtKeys

Extracts stories from the press agency NTB. The option is used like this:

```
Artkeys="SOS,ARB,KUL,POL"
```

The possible keys are: ARB, FRI, KRI, KRE, KUL, MED, NAT, KUR, POL, REL, SOS, SPO, ULY, UTD, VIT, VAR, OKO.Filtering can be used on this tag to extract stories based on the value in a key. The option is then used like this:

```
artkeys="<Class1>=<Value1>|<Value2>,<Class2>=<Value3>
<!--
 /* Font Definitions */
 @font-face
 {font-family:Verdana;
 panose-1:2 11 6 4 3 5 4 4 2 4;
 mso-font-charset:0;
 mso-generic-font-family:swiss;
 mso-font-pitch:variable;
 mso-font-signature:536871559 0 0 0 415 0;}
 /* Style Definitions */
 p.MsoNormal, li.MsoNormal, div.MsoNormal
 {mso-style-parent:"";
 margin:0in;
 margin-bottom:.0001pt;
 mso-pagination:widow-orphan;
 font-size:10.0pt;
 mso-bidi-font-size:12.0pt;
 font-family:Verdana;
 mso-fareast-font-family:"Times New Roman";
 mso-bidi-font-family:"Times New Roman";}
@page Section1
 {size:595.45pt 841.7pt;
 margin:.7in .7in 2.4in .7in;
 mso-header-margin:.5in;
 mso-footer-margin:.5in;
 mso-paper-source:0;}
div.Section1
 {page:Section1;}
 -->
```

Category

Determines the category from which stories should be retrieved. This option has no default but it **MUST** be set.
Option

```
Current= The category currently being viewed.
Current%
= The category currently being viewed with subcategories.
All
= All categories
%
= All categories
Categoryname
= A specific category
```

The category names available depends on your press agency feed.
The following are options for various press agencies.

- **AP**
  AP = The Associated Press (North America) http://www.ap.org. AP customers can use the following categories:
  AP% = All AP categories.
  APA = Nation
  APB = Nation/Special
  APC = Feature
  APD = Feature/Food
  APE = Feature/Entertainment
  APF = Business
  APG = Metro/Briefs
  API = Foreign
  APJ = Metro/Lottery
  APK = Op-Ed
  APL = Feature/Lifestyle
  APN = Metro
  APO = Weather
  APP = Nation/Politics
  APQ = Sports/Single
  APR = Sports/Racing
  APS = Sports
  APT = Feature/Travel
  APU = Metro
  APV = Advisory
  APW = Nation/Washington
  APX = Sports/Agate
- **CP**
  CP = Canadian Press (Canada) http://www.cp.org. CP customers can use the following categories:
  CPB = Business
  CPS = Sports
  CPN = National
  CPE = Entertainment
  CPP = Panamerican Games
  CPW = World
- **NTB** = Norsk Telegrambyr (Norway) http://www.ntb.no. NTB customers can use the following categories:
  NTBI = NTB Innenriks
  NTBU = NTB Utenriks
  NTBS = NTB Sport
  NTBO = NTB ¯konomi
- **RZ** = Ritzaus Bureau (Denmark) http://www.ritzau.dk. RZ customers can use the following categories:
  RZI = Ritzau Indland
  RZU = Ritzau Udland
  RZK = Ritzau Kriminal
  RZS = Ritzau Sport
  RZE = Ritzau Erhverv
  RZP = Ritzau Politik
- **TT** = Tidingarnas Telegrambyr AB (Sweden) http://www.tt.se/start/. TT customers can use the following categories:
  TTI = TT Inrikes
  TTU = TT Utrikes
  TTE = TT Ekonomi
  TTS = TT Sport
  TTP = TT Resultater ||

Count	Controls the number of stories that will be shown. By default, all stories modified today are shown. Enter any number between 1 and 999 in `count=x` to change the number of stories to be shown.

Keywords	If the news feed contains keywords (NTB and AP) you can extract stories based on keywords.To use keywords enter `Keyword=X` where X is the keyword. You can also do partial matching using the % sign like this: `Keyword=ol-%`. You can also do a partial match like this: `Keyword=%war%`. You cannot use multiple keywords.You can use
	`Category=AP[FP]`
	to display stories from both APF and APP. You can use multiple % signs so the string does not have to start with the keyword. You cannot use multiple keywords.The AP feed contains keywords for the state. To list all stories with the keyword *Alabama* you can use this tag:
	`<pbs:!agent category=AP% keyword=AL% cachetime=10>`

Leadin	Determines how many lead-in characters will be shown. By default, the entire lead-in is shown. Enter any number of characters from 0 to 9999 to show that number of lead-in characters. Setting `Visref=0` will turn off the lead-in option.
Location	Displays stories with a specific dateline. By default, this option is disabled. Entering `Location="X"` will show all stories with the specified dateline.The tag appears like this: `<pbs:!agent category=% location="New Delhi"cachetime=10>` Only stories listed in the specified dateline will be shown. You can further limit the output using the count option.
Maxpicturewidth	Determines the maximum width of the image.
Objectclass	Determines which object file should be used. By default, the general objects will be used. Object class can be set from 1 to 999 by entering `objectclass=x` where x is the number.
Priority	Shows stories matching a specific priority. By default, this option is disabled. If you enter `Priority=1` it will show all stories that have priority 1. You can specify only one priority for each tag. The priority codes differ from vendor to vendor, so some experimentation might be necessary.The tag then appears like this: `<pbs:!agent category=AP% priority=1 cachetime=10>`
Priosort	Sorts the story list by priority. By default, this option is disabled and stories are sorted by timestamp by default. Enter {{Priosort=1 }}to sort stories by priority. The result is that stories are sorted by priority code.
Start	Controls how many stories the tag should skip before it displays stories. This option can be used when you want to place an ad in the listing of stories or use multiple Newslist tags on the same page to control layout. By default, the system starts at the first story. You can set start story between 1 and 999 by entering the desired number in `start=x`

Theme	If the news feed contains a theme (AP) you can extract stories based on theme. By default this option is not used and all themes are listed.To use this option enter
	`Theme="X"`
	You can also do a partial match like this:
	`Theme="202%"`
	You cannot use multiple themes. A list of AP theme codes is available at www.support.us.saxotech.com in the FAQ section.Theme is used like this in the tag:
	`<pbs:!agent category=AP% theme="1000" cachetime=10>`
	The result is that only stories with a matching theme will be output.

Topstories	Displays a list of stories based on the newest priority document from your press agency. By default, this option is disabled. Entering `Topstories=X` will show all stories for the Top story category. A list of AP theme codes is available at www.support.us.saxotech.com in the FAQ section.The tag appears like this: `<pbs:!agent category=% topstories=9000 cachetime=10>` Only stories listed in the newest priority document will be shown. You can further limit the output using the count option.

Agent.pbo

<%Items%>	This field outputs the items.

AgentItem.pbo

Used to list stories.

<%Counter%>	The item counter. The counter starts at 1 for the first item.
<%\Date%>	The publication date preformatted.
<%DateHeader%>	The date header. The tag supports formatting, and can be used like this: `<%DateHeader$d="mmmm d."%>` It is only output when the date changes.
<%ImgLink%>	The image link. It must be used in an `<img>` tag.
<%IsoDateTime%>	The date and time in ISO format. It can be formatted using object script.
<%Leadin%>	The lead-in.

<%Link%>	The link to the entire story.Example use: `<a href="<%Link%>">Linktext</a>`
<%\Time%>	The publication time preformatted.
<%\Title%>	The title for the story.

pbs agentcategory

<pbs:agentcategory>

This tag can be used on all _**art.pbs** templates on which you display agency stories.

pbs history

<pbs:history>

The history tag is used to list older stories in the category being viewed. It can be used both on the section front and in a story. The function this tag provides has been superseded by the Newslist tag. It is only documented for historical purposes. The {{<pbs:history> }}tag might be used as follows:

```
<pbs:history useobjects=1 dateformat="mm/dd/yyyy">
```

Options

Bgcolor	Sets the background color on the title, when an object file is not used.
Bredde	Sets the width of the output in pixels or percentage when object files are not used.
Category	Determines the category from which the history is to be retrieved. By default, the current category is used. Select one of the following if another category option should be used: • **Current** - Stories from the current category and subcategories. • **Category** - Stories from a single category. • **Category%** - Stories from a single category with subcategories. • **%** - All categories.
Color	Sets the color on the title above the links, if an object file is not used.
Count	Determines how many stories the history tag should display. By default, the value is set in the category register. Enter `Count=#` to override the setting in the category register.
Dateformat	Controls how the date will be shown by the `<%\date%>` tag. Note that it is also possible to use the `<%isodate%>` variables and format dates yourself. The default date format is dd/mm. This option can use other standard date format options.
Fontsize	Sets the font size of text when object files are not used.
Ledetekst	Sets the title when an object file is not used.
Objectclass	Determines which object files the system should use. By default, this option is disabled. By setting `Objectclass=1` the system will look for object files named History1.pbo and HistoryItem1.pbo. By using other numbers, the system will look for object files named History#.pbo where # is the number.
Object files	Enter `Useobjects=1` to use object files. By default, object files are not used.
Useobjects	This option controls the use of object files. By default, this option is disabled. Enter {{Useobjects=1 }}to make the system look for object files.

***Visref ***	Determines how many characters of the summary the <%text%> variable should use. By default, it uses 0 characters. Enter `visref=x` where x is any number between 0 and 9999.

History.pbo

The wrapper for the objects.

<%Items%>	The position where the code for each item will be output.

HistoryItem.pbo

The file HistoryItem.pbo is output for each item in the history list.

<%\Date%>	The story's date. Formatting is controlled by the **Dateformat** option.
<%Link%>	The link to the story.
<%Text%>	The summary of the story. The amount of text shown is determined by the *Leadin *option.
<%\Title%>	The title of the story.

pbs keywordindex

<pbs:keywordindex>

The index tag is used to build keyword indexes. The stories must be tagged with keywords to appear in the index.The tag must be placed on the template **gen_idx.pbs**. Stories must be tagged with one or multiple keywords to appear in the index. To link to the page, you must link to* /apps/pbcs.dll/index*. To specify a category, add **?Category=Categoryid**.

Options

Current	The keyword index only show stories from the category currently being viewed. This requires that the category is specified in the URL.
Descending	This option reverses the sort order of the index. By default, the index is sorted ascending.
Includesubs	The keyword index shows stories from the category currently being viewed and subcategories.
Objectclass	Determines which object class will be used.

Range	Controls which items are output. Range is set by using the option like this:
	`Range="0-9"`
	will output all entries with numerical keywords.
	`Range="0-Z"`
	will output all entries.
	`Range="A-Z"`
	will output all entries with alphabetic keywords.
	`Range="A-A"`
	will output all entries with keywords that begin with A.
	The default is A to Z.

Useobjects	This tag will use object files if they exist. To turn off the use of objects, enter `Useobjects=0`

KeywordIndexHeader.pbo

<%Header%>	The header. The header is the first letter of the alphabet.
<%TitleList%>	This field outputs the items.

KeywordIndexLink.pbo

<%ArticleLink%>	The link to the story.
<%ArticleTitle%>	The title of the story.
<%ArticleDate%>	The date the story was published.
<%IsoArticleDate%>	The date in ISO format. This date can be formatted using the variable formatting options.

KeywordIndexTitle.pbo

<%LinkList%>	This field outputs the items.
<%\Title%>	The keyword.

pbs mostread

<pbs:mostread>

Used to list the most frequently read stories for a category. Please see LiveArtStatUpdate on how to enable the use of `<pbs:mostread>`.

Mostread does not support cachetime=xx so do not insert ! in this tag. It will run uncached.

Templates

This tag can be used on all templates.

Documentation of the "All" parameter was wrong, and documentation of parameter "includenodate" was missing.

Options

All	Enter this option to omit the date rage filter. `<pbs:mostread count=10 all>`
IncludeNodate	Set this parameter to 1 (it is default 0) to include undated stories. `<pbs:mostread count=10 all includenodate=1>`

Category	Determines the category for which stories will be listed. By default, stories are listed for the current category. To list popular stories for all categories, use the option
	`Category="%"`
	You can also specify a single category by using
	`Category="NEWS"`
	To list a category with subcategories, use the option like this:
	`category="NEWS%"`

Count	Determines how many stories will be shown. By default, 10 stories are shown. Enter `Count=X`, to show X stories.
CurrentDate	Lists only the most read stories with today's date as the publication date. Note that the category must be set. For example: `<pbs:mostread showref=0 count=10 CurrentDate category=%>`
Days	Limits the age of the stories shown. This option uses the publication date of the story and counts backwards. The Days option is faster than the StatDays option. Please see **StatDays**. By default, stories published in the last 100 days will be shown. Enter `Days=X` stories to show stories published the last X last days. The value can be set from 1 to 365. Days does not work with **All** and specifying both options will make the tag ignore **All**.
LastMonth	Lists the most read stories from the last month. For example, if the current date is May 5th, this option will list the most read stories from April 1st to April 30th.Best practice: If you use LastMonth, do not use LastWeek.
LastWeek	Lists the most read stories from the last week (Monday to Sunday). Best practice: If you use LastWeek, do not use LastMonth.
Minread	Sets the minimum of hits a story must have before it will be listed. By default, a story must have 5 views before it is listed. Enter `minread=x` where x is any number between 1 and 99.
MostEmailedArticles	Tracks the number of times that a story has been e-mailed (Tip a Friend). This option lists the stories that have been e-mailed most times on a chosen date.Best practice: If you use MostPrintedArticles, do not use MostEmailedArticles.

MostPrintedArticles	Tracks the number of times that a story has been printed. Update of the number of stories printed requires use of the PrintArt.pbo or that an additional "print=1" parameter be appended to the URL of the "Print article..." link. This option lists the stories that have been printed the most on a chosen date. Best practice: If you use MostEmailedArticles, do not use MostPrintedArticles.
Objectclass	Sets the object class for the tag. By default, object class is not used. Enter `objectclass=x` where x is any number between 1 and 999 to set the object class.
Profile	Displays the most read stories based on profile. By default, this option is disabled. Enter `Profile=X` to show stories with the profile X.
Relprofile	This function makes the system look for the most relevant profile for a story. If you specify the profile in the **MostRead** tag, the link to the story will contain the profile ID with the most matching points for the story. This function is used when your design is attached to the profile and not the category, or you have a **MostRead** tag on the story page showing stories from the same profile.
Showref	Determines how many characters of the summary will be shown. By default, no characters from the summary will be shown. Enter `Showref=150` to show the first 150 characters of the summary.
Site	Shows the most popular stories from another site. By default, this option is disabled. Enter {{Site=X }}in the tag to show stories from the site with site code X.
StatDays	Limits the age of the stories shown. StatDays uses the date for the last time the story was read which is the last time statistics changed on the story.For example, a story about Tasmania was published 100 days ago. This story might be read many times today, if a girl from Tasmania was married to a Danish prince yesterday. This story would not be shown with the Days option set to 7, but it would be counted if StatDays was set to 7. Please see **Days**.

MostRead.pbo

<%ItemCount%>	The number of stories.
<%Items%>	Lists the items.

MostReadItem.pbo

<%\Date%>	The date for the story.
<%ExtLink%>	The link to the story when you use the site option.
<%IsLastItem%>	This field has the value 1 when the last story in the list is shown.
<%ISODate%>	The date in ISO format.
<%Leadin%>	The summary for the story. This field only has a value if showref is larger than zero.
<%Link%>	The link to the story.
<%Position%>	The position number for the story.
<%\Title%>	The title of the story.

pbs newslist

<pbs:newslist>

The **Newslist** tag is used to build overviews of stories. This tag can be used on all templates.

Options

AccessDomain	Used in ClickShare for setting the common domain for clicks.
Cachetime	The option cachetime is used to determine how long a Newslist should be cached. The function is typically used on front pages and for breaking news lists.By default, cachetime is disabled. To use the cachetime option, you must run the tag uncached. The option is used like this: `Cachetime=X` where X is the number of minutes the list should be cached.

Noduplicates	This functionality allows a synchronization of `<pbs:newslist>` tags. By incorporating the NoDuplicates and SynchID parameters, the developer of a website is able to control whether or not stories are allowed to repeat, relative to two or more `<pbs:newslist>` tags. This enables easy removal of duplicate teasers in newslists. Note that `<pbs:newslist>` tags which you would like to be synchronized must be placed in the same file. By default, this attribute is disabled. Enter `noduplicates=1`, for the PBCSDupArtFilter.dll to maintain a list of stories to be displayed. For more information on this dll file, please see PBCSDupArtFilter. Setting the synch ID value to a specified number effectively groups all `<pbs:newslist>` tags that have the same synchronization ID.The following code provides an example of how to use the SynchID and NoDuplicates parameters:`<pbs:newslist days=20 multisite=1 noduplicates=1 site="ap,cj" synchid=1 count=4 objectclass=1><pbs:newslist days=20 multisite=1 noduplicates=1 site="ap,cj" synchid=1 count=2 objectclass=1>`
	
	In the example above, both `<pbs:newslist>` tags share a common **SynchID** value, which allows SAXOTECH Online to group these tags together. By setting the **NoDuplicates** parameter, the PBCSDupArtFilter.dll will potentially incorporate the results of each `<pbs:newslist>` into its list of stories. If a story is found to be repeated, the dll will not add it to the list. As a result, it will present a list of non-repeating stories. For more information on this dll file, please see PBCSDupArtFilter.
	
	When using NoDuplicates and the same SynchID on several newslists in the same file you have to make sure that unique cachefiles are generated
	
	Example:
	
	`<pbs:!newslist cachetime=2 category="NEWS" count=3 days=5 useobjects=1 objectclass=1 noduplicates=1 synchid=1>`
	`<pbs:!newslist cachetime=2 category="NEWS" count=3 days=5 useobjects=1 objectclass=1 noduplicates=1 synchid=1>`
	
	These two newslist tags are 100% identical, this will generate the same cachefile for the two tags. Effectively this means that the second newslist tag overwrites the cachefile from the first tag as the system builds cachefiles based on the parameters in a newslist tag.
	
	What you should do to prevent this from happening is the following:
	
	`<pbs:!newslist cachetime=2 category="NEWS" count=3 days=5 useobjects=1 objectclass=1 noduplicates=1 synchid=1 uniquecache=1>`
	`<pbs:!newslist cachetime=2 category="NEWS" count=3 days=5 useobjects=1 objectclass=1 noduplicates=1 synchid=1 uniquecache=2>`
	
	There is added `uniquecache=X` as a parameter in the tag. Now the cachefiles will be different, and the content will be correct.
	
	So a new parameter "uniquecache" has to be added to the tag if they are identical. We strongly recommend that you always add that parameter when using noduplicates and synchid, no matter if they are the same, or different.

Objectclass	Controls which object file will be used. Enter `Objectclass=1` to make the system look for News1.pbo and NewsItem1.pbo. The rules defining which object files will be used are described in the section About Object Files. You should also read the section on the folders in which the system will look for object files, which can be found in Directory structure.
SynchID	Setting this value to a specified number effectively groups all `<pbs:newslist>` tags that have the same synchronization ID. See **NoDuplicates**.
Useobjects	Determines whether object files should be used. If you enter `Useobjects=1`, the system will look for an object file to define the layout. The object files are described later in this chapter.

Output options

The newslist tag has the following output options. These options control which stories will be shown by the tag.

Categories	Used when you want to select multiple categories for the Newslist and it is not possible to use the pattern matching that the category option supports. The category option is faster and should be used if possible. This option is only used when the option category cannot be used.To select multiple categories, enter them into the tag. For example,
	`Categories="NEWS,SPORTS"`
	selects stories from both the *News* and *Sports* categories as in this example:
	`Categories="news%,sports0[1-3],arts[^4],opinion"`
	The options are:
	`Categories="news%,sports%"`
	Selects stories from all News and Sports categories
	`Categories="news0[5-6],sports0[1-2]"`
	Selects stories from the categories news05, news06, sports01 and sports02. As with the category option, the range used must be one character. Writing `Categories="CATEGORYID05-06"` is not supported.
	`Categories="news0[^5],sport0[^1]"`
	Selects stories from all categories except for news05 and sports01.

ExcludeCategories
From versions 6.4.0sr1
	Used when you want to exclude categories for the Newslist. To exclude multiple categories, enter them into the tag. For example,
	`ExcludeCategories="NEWS,SPORTS"`
	excludes stories from both the *News* and *Sports* categories as in this example:
	`ExcludeCategories="news%,sports0[1-3],arts[^4],opinion"`
	The options are:
	`ExcludeCategories="news%,sports%"`
	Excludes stories from all News and Sports categories
	`ExcludeCategories="news0[5-6],sports0[1-2]"`
	Excludes stories from the categories news05, news06, sports01 and sports02. As with the category option, the range used must be one character. Writing `ExcludeCategories="CATEGORYID05-06"` is not supported.

Channel	This option uses the **Channel** property from the editing interface to control which stories the tag should list. This option is typically used when you build a page that is to be used for either PDA or WAP distribution.The default is to output stories through the Web channel. The options for channel are: Web, PDA or WAP.The tag would appear as follows: {{<pbs:newslist channel=PDA> }}and the resulting output would be that only stoies with a matching channel will be output when this option is used.
Count	This option controls how many stories should be shown. The default is 5 stories. Enter from 1 to 999 stories to change the option. When you use profiles, the cache rows option on the profile will set the hard limit for how many stories the tag can show.
CounterStart	Determines the counter parameter's offset. Enter counterstart=# where # is the number used for the offset from the counter setting.

Createdby
From version 6.2.0
	Used when you want to select stories from specific users.
	
	Example:
	
	`<pbs:newslist createdby="UserId-3125617,UserId-3125122,UserId-111111">`
	
	The tag will select only stories submitted by those users.
	
	Example:
	
	`<pbs:newslist createdby="#currentuser#">`
	
	The tag will retrieve current logged in user id from the session object and list stories.

Date	This function allows you to pick stories from one date. By default, this is not used. By setting Date="99999999" you can select stories without a date. You can use any date in YYYYMMDD format to select stories from only that date. Example: `<pbs:newslist date="99999999">`
Days	Controls how many days back the system will look for stories. The default is 7 days. Enter from 1 to 999 days to change the option. For the sake of speed, you should never use more days than necessary. Setting this option to a high number will slow down the page.When you use profiles, the Week span option on the profile will set the hard limit for how far back the system looks for stories. If you have a week span of 7, Days=49 is the highest number of days that will have any meaning.
Daystart	This option controls how many days the tag should skip relative to today's date before it starts displaying stories. Setting Daystart=1 and Days=2 lists stories from yesterday. You cannot modify the date using Allow Date Override (Ado=1).
Even	Enter `Even` in the tag to output only items with an even position code. By default this option is not used. Enter `Even` in the tag to output only items 2, 4, 6, 8, and so on.
ExactTaxonomyMatch	This settings controls whether the entire taxonomy tree is used when using Taxonomywords parameters. By default the entire taxonomy tree is used. This means that stories which have the taxonomy word identified in the newslist including the parent structure will be selected. If this settings is set to 1 then only stories which have specified taxonomy ids assigned to the directly will be selected.
FilmID	This option is like Location. It is used for selecting stories with a keyword of the **filmid** instead of location for Location.

Keywords	This option lets you select stories based on the keywords with which they are tagged. By default, this option is not used.You can use
	
	`keywords="politics,Cuba"`
	
	to select stories that have been tagged with both keywords. If your keyword has spaces, you must write the keyword with underscores. The option must be written like this:
	
	`keywords="keyword_with_spaces"`
	
	To select stories that have been tagged with the name of the category being viewed, use the keyword #category#. If you are viewing a subcategory and want to use the name of the main category as a keyword, use the keyword #maincategory#.

Kwnot	This option separates the keywords with NOT. The default option for keywords is to use the AND separator. Entering `Kwnot=1` in the tag determines that the keywords will be separated with NOT.

Kwor	This option separates the keywords with OR. The default option for keywords is to use the AND separator. Entering `Kwor=1` in the tag determines that the keywords will be separated with OR.
Lastdate	Makes the tag only select stories from the last publication date. This will override the count and days options.
Letter	Used with Alphasort to select which letters should be output. The system looks for stories with a Net title that matches the designated letter. If the story does not have a Net title, the story title will be used instead.This option has no default setting. There are several methods to use this option. • **Letter D** - Outputs all stories that start with the letter D. • **Letter=A-G** - Outputs all stories that start with letters from A to G. • **Letter=0-9** - Outputs all stories that start with numerals.
Location	Allows you to select stories based on the story's **Location** property. The text specified in this option must be an exact match for the value from the **Location** property on the story. By default, this option is not used. Enter `Location="XXX"` to make the tag select stories with XXX in the **Location** field.
Multisite	Allows the tag to retrieve stories from multiple site codes. Used with **Site. *Only use *Multisite** if articles are retrieved from multiple sites. The following code provides an example of how to use the Multisite parameter (note that articles are retrieved from two sites): `<pbs:newslist multisite=1 site="ap,cj">` By default this option is not used. If you enter Multisite in the tag, you can use the site option to select stories from multiple sites.
Odd	Enter `Odd` in the tag to output only items with an odd position code. By default this option is not used. Enter `Odd` in the tag to output only items 1, 3, 5, 7, and so on.
Prio	Used to select stories on one priority only. Setting Prio=1 will select all stories with priority set to 1. To select stories without a defined priority, use the option Prio=0 in the tag. Using Prio disables the Priosort setting.
Prioonly	Only stories with a defined priority will be shown. If you want to show stories without a defined priority, use the option Prio=0. If you use Prioonly, you must use Priosort too.
Priosort	Sorts the story list by priority. By default, this option is disabled and stories are sorted by timestamp. Enter {{Priosort=1 }}to sort stories by priority. The result is that stories are sorted by priority code.

Profile	Used to select stories based on profiles instead of categories. A profile is a collection of taxonomy words. For more information please see Taxonomy and profiles. By default,
	`<pbs:newslist profile>`
	takes the profile from the URL. For example, profile=1038. When you place a profile in the **Newslist** tag, the profile selected on the **Category** settings page in the edit interface will be used. You can override this profile by entering
	`&Profile=XXX`
	in the URL, where XXX is the profile number. You can specify the profile directly by entering
	`Profile=XXX`
	in the tag. This will lock the tag to display stories matching profile number XXX. You can limit the selection of stories using categories with profiles. This means that stories matching a profile also will have to match the category.

ProfileLink	Used to turn off the profile in the link to the story. By default, the profile link is output. By setting {{Profilelink=0 }}in the tag, the profile will not be output.

ProfileSite	Determines from which site the profile should be selected. Enter `ProfileSite=xx` where xx is the site code for that site.
ProfileType	Determines the sub-profile to select. Enter `ProfileType=xxx` where xxx is the Profile Type ID.

Refant	Controls how many stories should have a lead-in. By default, all stories have a lead-in. Enter
	`refant=x`
	where x is any number from 1 to 99 to display lead-ins on the specified number of stories.Saxotech strongly recommends using objectscripting instead of the refant option to control how many stories should have a lead-in. By using objectscript, you can keep the number of object files low while still displaying a lead-in. For example:
	`NewsitemX.pbo`
	`[%{Counter=1|counter=2}`
	`<%summary%>`
	`%]`

Relprofile	This function makes the system look for the most relevant profile for a story. If you specify the profile in the Newslist tag, the link to the story will contain the profile ID with the most matching points for the story. This function is used when your design is attached to the profile and not the category, or you have a Newslist tag on the story page showing stories from the same profile.By default, the profile specified in the tag is used. By adding relprofile to the tag, the most relevant profile will be used in the link. **Relprofile logic** It uses the sorting functionality in this order: 1. Profile priority 2. Then Profile level (It takes the deepest profile in the profile tree) 3. Then Taxonomy level (It takes the deepest Taxonomy in the Taxonomy tree)
Reltaxonomy	This function enables you to show a list of related stories for each element in a newslist. This solution is based on taxonomy, where taxonomy is used to group stories so that they are shown together. The related stories will always be sorted by the Last Changed date. The function is used with the **RelStoriesItems.pbo** and **RelStoriesItem.pbo** object files.Note that this feature is hard on the database, so limited use is advised. The **Or** operator gives fewer database queries (because only one taxonomy word is used for the related taxonomy) so it should be used whenever possible.By default, the reltaxonomy option is disabled. Add Reltaxonomy=1 to the tag to enable the related taxonomy feature.
Reltaxonomycount	The number of related stories to show in the list under each element. By default, five stories are shown in each list. You can enter any number up to (2^31^)-1.
Reltaxonomyoperator	The **Or** operator gives fewer database queries (because only one taxonomy word is used for the related taxonomy) so it should be used whenever possible. By default a single taxonomy word is used for related taxonomy.To use reltaxonomyoperator enter Reltaxonomyoperator=And - Several taxonomy words are used for related taxonomy.
Reltaxonomytopnode	Must be set when using *Reltaxonomy. *If no topnode is wanted, this should be set to the root node (ID=0). This is not advised as the system will only match on the 5 taxonomy words with the lowest ID.Instead, a subtree should be created in the taxonomy tree, which is used to show the list of related stories. The first node of the subtree should be used as the Topnode. If the topnode is not set, the system will not show related stories and an exception will be raised. The option would appear as follows: `reltaxonomytopnode=[TaxonomyID]`
ShowBloggerInfo **From versions 6.2.0sr20, 6.3.0sr3**	Determines if blogger fields such as **CreatedByUserFirstName** is to be outputted (=1) or not(=0). Default is 0.

Site	Determines the sites from which the tag can retrieve stories. This option is used with *multisite.*By default, this option is not used. If you enter `Site="TB,SB,GJ"` in the tag, you can select stories from those three site codes.
Start	This option controls how many stories the tag should skip before it displays stories. This option can be used when you want to place an ad in the listing of stories or use multiple Newslist tags on the same page to control layout.By default the system starts at the first story but you can set the option to start at any story number between 1 and 999.
Status	This option makes it possible to show stories depending on their status. To enable this, it must be activated in the Publicus.ini by setting the following: `AllowChangingPublishedStatusOnNewslistTags=1` in the GENERAL section of the publicus.ini. This enables the status parameter on the Newslist tag. This parameter must be set to a set of valid statuses separated by comma. The valid statuses are: • 0: Pending • 1: Unpublished • 7: Restricted • 8: Archived • 9: Published • 99: Backed up • 10: Locked For example: `<pbs:newslist days=2 status="1,9,3,4">` It is important to use quotes (") around the status string. Otherwise, it only uses the first element in the list. Use caution when using the category or profile from the URL in the tag (#category#) because this may allow readers to find unpublished stories from other categories and/or profiles.
Taxincl or Taxinclude	This option is used the same way as **Profile**. See **Profile** for more information.
Taxonomywords	You can output stories matching specific taxonomy words. By default, this option is not used. Use `Taxonomywords="12,14"` in the Newslist tag. When you specify multiple words, they will be separated by AND. By adding `Kwor=1` to the tag, the terms will be separated by OR.Example:`<pbs:newslist taxonomywords="12,14" kwor=1>`. Please read ExactTaxonomyMatch.
Themes	Allows you select stories based on theme. By default this option is not used. Using this option allows you to select stories using multiple themes. The option is used like this: `themes="computers,telephone,internet"`

Sort options

By not using any sort options, stories will be sorted this way:

- The default sort order when profiles are not used is date, modified time, Category ID and Article ID with lowest Article ID first.
- The default sort order when profiles are used is date, priority and Article ID with highest Article ID first.
- When you combine the sort options, Alphasort will take precedence over Priosort and Datesort. Datesort takes precedence over Priosort when these two options are used together.
- The following options for sort are available:

Alphasort	Sorts the stories alphabetically. Only the first letter of the title will be used. By default, this option is not used. Enter `Alphasort=1` to sort the stories alphabetically by the first letter of the title.

Datesort	Sorts the stories by the following criteria in the listed order: • By the Publication date or the Publish Control From date (whichever is most recent) with the newest story first. • By either the Last Modified time or the Publish Control From time (whichever is most recent)Using Datesort disables Priosort. By default, this option is not used. Enter `Datesort=1` in the tag to sort the stories by publication date.

Extrasortfield	When **ExtraFieldDef** is used in Publicus.ini, the Newslist tag can be sorted by the extra fields set up in that section. Please see Extra fields definition. By default, this option is not used.
	`ExtraSortField=1`
	enables sorting by extra fields. The sorting is done on the FIRST field in the fields parameter. The fields are taken from the fields defined for EditExtraFields in the Extra fields definition section in the **Publicus.ini**. For example:
	`<pbs:newslist category="MOVIE" count=10 useobjects=1 objectclass=1 profilelink=0 ExtraSortField=1 ExtraSortFieldOrder=asc fields=movie_titel,movie_manuscript>`

Extrasortfieldorder	Sets up a sort order for the ExtraSortField. Legal values are **asc** and **desc**. By default, the sort order is ascending. `ExtraSortFieldOrder=desc` sets up a descending sort order for the ExtraSortField.

Letter	Used with Alphasort to select which letters should be output. The system looks for stories with a Net title that matches the designated letter. If the story does not have a Net title, the story title will be used instead. By default this option is not used. Entering
	`Letter=D`
	will output all stories that start with the letter D. Entering
	`Letter=A-G`
	will output all stories that start with letters from A to G. Entering
	`Letter=0-9`
	will output all stories that start with numerals.

Priosort	Sorts the stories by the following criteria in the listed order: 1. By either the Publication date or the Publish Control From date (whichever is most recent) 2. By priority 3. By either the Last Modified time or the Publish Control From time (whichever is most recent)Web guests will see today's prio1-n stories, yesterdays prio1-n stories, etcetera. If you want to sort stories by priority and then by date, use **Priosortonly**. By default this option is not used. Enter `Priosort=1` to sort the stories by date and then by priority.

Priosortonly	Sorts the stories by the following criteria in the listed order: 1. By priority 2. By either the Publication date or the Publish Control From date (whichever is most recent) 3. By either the Last Modified time or the Publish Control From time (whichever is most recent) Example: `<pbs:newslist priosortonly>`
Ratingsort	Sorts the stories by rating. By default this option is not used. Enter `Ratingsort=1` to sort the stories by rating.

Formatting options

These options control some of the formatting of the content output by the system and determine whether related content should be shown.

Alltext	Makes the system output all the text in the first paragraph. The variable `<%text%>` is used for displaying the story.
Dateformat	Sets the format for the preformatted date field. The default setting for this option is language dependent. The system will output the dates in the normal format for your language.Format the dates by using `Dateformat="mmmm dd, yyyy"` or any other date format option. You can also format the date field directly in the object file.Please see Formatting dates for more information.
ExternalLinks	Used to enable links to external stories related to the story currently being viewed.By default, this option is not used. Enter `ExternalLinks=1` to show links to related stories outside your website.With the parameter enabled the system will look for and act on the <%ExternalLinks%> tag in the NewsItem.pbo. That tag calls the NewsItemExternalLinks.pbo file. The NewsItemExternalLinks.pbo will recognize the tag <%NewsItemExternalLinkItem%> that uses the pbo by the same name.
Fields	Specifies which extra fields should be available in the Newslist. The fields are specified in a comma separated list like this: `Fields="Edition,Section,Tearsheet"`
Image	Determines whether images will be shown in the Newslist. Enter Image=1 to show images in the newslist.
InternalLinks	Used to enable links that list stories related to the story currently being viewed. By default, this option is not used. Enter `InternalLinks=1` to show links to related stories.
KeywordLinks	Enables links to stories with the same keywords as the story currently being viewed. By default, this option is not used. Enter `KeywordLinks=1` to show links to stories with the same keywords.
KeywordLinksCount	This option tells the system how many stories with the same keywords should be listed. By default 5 stories are shown. Enter `KeywordLinksCount=2` to show two stories, for example.
KeywordLinksMonths	This option controls how many months back the system should look for stories with the same theme. By default the system looks back 3 months. Enter `KeywordLinksMonths=6` if the system should look back 6 months, for example.
Leadin	Controls how many characters of the lead-in should be displayed. If there is no lead-in, the first paragraph will be used. By default there is no lead-in.Enter any number between -1 and 9999 to change the number of characters displayed in the lead-in. Entering -1 will result in the system displaying the entire first paragraph of the story.
Multimedia	Enables direct links to multimedia objects connected to a story. By default Multimedia links are not output. Enter `Multimedia=1 }}to show{{ }}links to multimedia objects when the field {{<%MultiMediaItem%>` is present in the object file.

ProfileHeader	Determines whether a profile header will be shown for the newslist. Enter `ProfileHeader=1` to show a profile header for the newslist.
Showref	Determines how many characters of the summary will be shown. By default, no characters from the summary are shown. Enter Showref=# where # is the number of characters from the summary to be shown.
Source	Determines whether the source will be displayed in the newslist. Enter `Source=1` to display source in the newslist.
Teaser	This option is used for turning off the use of teasers on stories. By default, the teser registered on the story will be used. Enter `Teaser=0` to disable the teaser. The standard title and summary will be used.
ThemeLinks	Enables links to stories with the same theme as the story currently being viewed. By default, this option is not used. Enter `Themelinks=1` to show links to stories with the same theme.
ThemeLinksCount	This option tells the system how many stories with the same theme should be listed. By default, 5 stories are shown. Enter `Themelinkscount=2` to show two stories, for example.
ThemeLinksMonths	This option controls how many months back the system should look for stories with the same theme. By default, the system looks back 3 months. Enter `ThemeLinksMonths=6` if the system should look back 6 months, for example.
Timeformat	Sets the format for the `<%\date%>` parameter in the object file. You can also format the time field directly in the object file. By default, time is displayed as hh:mm using a 24 hour clock. Please see Formatting dates for more options.

Special options

Allow date override (Ado)	Used to allow the date to be overridden in the URL. This option must be set if you want to use the **Frontpagearchive** function. Ado is ***not*** to be used with Daystart. By default, this option is not used. Enter `Ado=1` to override the date used by the Newslist tag.
Frontpageflag	Makes the Newslist tag on the front page link to the template **gen_fart.pbs** instead of **gen_art.pbs**. Please see Template mapping. By default, this option is not used. Enter `Frontpageflag=1` to use gen_fart.pbs.
Frontpageflagforce	This makes the Newslist tag on all templates link to the template **gen_fart.pbs** instead of **gen_art.pbs**. By default, this option is not used. Enter `FrontpageflagForce=1` in the tag to use gen_fart.pbs.

There are multiple object files that can be used by the Newslist tag. Please see About Object Files and Position-based design. This section lists the base names.

News.pbo

This file contains the main framework for the tag. It is possible to use the condition tags to exclude the whole file from being shown.

<%Items%>	This tag outputs all the items.

NewsItem.pbo

The following list of parameters is available for use in the object file.

<%ArticleStatus%> **From version 6.2.0**	Contains the status of the story. See Article.pbo for available status. This is only true when the 'ignorestatus' parameter is used on the newslist tag.
<%ArtID%>	The article ID for shorter links, which can be used without category and date, as here: `http://www.yoursite.com/apps/pbcs.dll/article?AID=9999607210374`. This field is u for cellphone payment for stories.
<%ArtIDLink%>	The complete story ID. It is for use with cellphone payment of stories.
<%AvgRanking%> **From version 6.4.0**	Contains the average ranking of a story.
<%BlogEntry%> **From version 6.2.0**	Contains whether the story is a user generated story (blog entry) (=1) or not (=0)
<%Byline%>	The byline on the story.
<%CaptionBody%>	The main text for the picture.
<%CaptionByline%>	The byline for the photo.
<%CaptionHead%>	The heading for the picture.
<%Category%>	The category ID for the category in which the story is shown.
<%CategoryLink%>	The link to the category. This field is often used in the following link: `<a href="<%categorylink%>"><%Categoryname%></a>`
<%Categoryname%>	The name of the category in which the story is placed.

<%Counter%>	This field starts with the value 1 and is increased by one for each story.
	`[%{counter=1}Article 1%]`
	`View ad 1`
	`[%{counter=2|counter=3}Article 2 and 3%]`
	Melt our newslist tags together with this function and increase database connections.

<%CreatedByUserId%> **From version 6.2.0**	Contains the creator's user ID if the story is a blog entry.
<%CreatedByUserFirstName%> **From version 6.2.0**	Contains the creator's first name if the story is a blog entry.
<%CreatedByUserSurName%> **From version 6.2.0**	Contains the creator's last name if the story is a blog entry.
<%CreatedByUserEmailAdr%> **From version 6.2.0**	Contains the creator's email address if the story a blog entry.
<%Date%>	The date the story was last modified. This field uses the **Dateformat** option in the tag for formatting.
<%DateHeader%>	This field outputs the date, but only when the date changes in the listing. This field uses the **Datefor** option in the tag for formatting.
<%Dateline%>	The dateline from the story.
<%DesignClass%>	The value of the design class.
<%DesignClassChanged%>	The value 1 when the design class changes in the list of stories. It is written:`<%DesignClassChang`
<%ExternalLinks%>	Used to show links to external objects that are linked to the story. For more information please see NewsItemExternalLinks.pbo

<%Field_xxx%>	This field is used for extracting extra fields from the story. Replace xxx with the name of the field. Th fields must be specified in the tag. The maximum number of characters that will be output for extra fi is 255.
<%ForumLink%>	Used to link to the forum on the story.
<%Heading2%>	This is the subtitle for the story.
<%ID%>	The article ID, which can be used with category and date as here: `http://www.yoursite.com/apps/pbcs.dll/article?AID=/20060721/PCSPORT/607210`
<%ImageHeight%>	The height of the image.
<%Imagelink%>	The link to the main story photo. You use the field like this: `[%{Imagelink}<img src="<%Imagelink%>&MaxW=120&MaxH=80" border=0 hspace=1 vspace=1 alt="" align="right">%]` The options for setting the maximum width and height of the image have also been added to the link can use either one or both.
<%ImageOrientation%>	The image orientation. The field has the value P for portrait and L for landscape. It is written: `<%ImageOrientation%>`
<%ImageRef%> **From version 6.2.0**	The reference to the image displayed. It is useful for zooming and required for zooming on teaser im
<%ImageWidth%>	The width of the image.

Image zoom	The Newslist tag does not support the zoom function you have on the story page. It is possible to bui this function using the link field. You do it one of these ways:
	`<a href="<%pbcs.dll%>/misc?url=/templates/zoom.pbs&<%Imagelink$c(23)%>" >Zoom</a>`
	`<a href="<%pbcs.dll%>/misc?url=/templates/zoom.pbs&aid=<%ArtID%>" >Zoom</a>`
	`<a href= "<%pbcs.dll%>/misc?url=/templates/zoom.pbs&dato=<%IsoCreateDateTime$d(" yyyymmdd")%>&kategori=<%Category%>&lopenr=<%id%>&ref=<%ImageRef%>" >Zoom</a>`
	The latter zoom link is the only one that works with Teaser images.

<%ingres%>	**This field outputs** the summary field. It will not show parts of the first paragraph if a summary does exist. This field is used with the option **Alltext.**
<%InternalLinks%>	Used to show links to internal objects that are linked to the story. For more information please see NewsItemInternalLinks.pbo.
<%IsCurrent%>	This field has the value 1 if the story is currently being viewed. This function can only be used when use the Newslist tag on a story page. This function can be used to make the current story bold in the of stories.
<%IsLastItem%>	This field has the value 1 when the last story in the list is shown.
<%IsoCreateDateTime%>	The creation date and time in ISO format. (YYYYMMDDHHMMSS). For more information on formatti please see <%ISOpubdate%>. This field can be formatted using object script. It is written: `<%IsoCreateDateTime%>`
<%IsoDateTime%>	The modification date and time in ISO format. (YYYYMMDDHHMMSS). For more information on formatting, please see <%ISOpubdate%>. This field can be formatted using object script.
<%IsoPubDate%>	The publication date in ISO format. (YYYYMMDD). This variable should be used in conjunction with formatting functions. Please read the section Formatting dates. This field can be formatted using obj script.

<%IsoPublishFromDateTime%>	The publishing date and time in ISO format. (YYYYMMDDHHMMSS). For more information on forma please see <%ISOpubdate%>. This field can be formatted using object script. It is written: `<%IsoPublishFromDateTime%>`
<%IsoPubDateTime%>	The publication date and time in ISO format (YYYYMMDDHHMMSS). This field can be formatted usi object script. It is written: `<%IsoPubDateTime%>`
<%KeywordLinks%>	Outputs stories that have the same keywords as the current story. The function uses the object files NewsItemKeywordLinks.pbo and NewsItemKeywordLinksItem.pbo.
<%Link%>	The URL to the story.
<%MainCategoryLink%>	The link to the main category if you are listing an entry from a subcategory. This field is often used in following link:{{<a href="<%MainCategorylink%>"><%MainCategoryname%>}}If the story is plac the main category, this variable will contain no value.
<%MainCategoryname%>	The name of the main category for the category in which the story is placed if the story is placed in a subcategory. If the story is placed in the main category, this variable will contain no value. It is writter `<%MainCategoryname%>`
<%MainProfile%> **From version 6.2.0**	Contains the profile chosen by the user in the edit interface as main profile when using multiprofile. tl setting OnlyUseEmbeddedNewsList in publicus.ini must be set to 1.
<%MultiMediaItem%>	Used to link directly to a multimedia object that is attached to the story. It targets the object with direc enabled. This field uses the object file NewsItemMultiMedia[Extension].pbo. [Extension] is replaced b media type extension.
<%Prio%>	The priority given to the story.
<%Profileid%>	The profile ID for the item.
<%ProfileName%>	The name of the profile when profiles are used.
<%PubDate%>	The publication date. This field uses the date **format** option in the tag for formatting.
<%PubDateTime%>	The publication date and time. This field uses the **Dateformat** and **Timeformat** options in the tag for formatting.
<%PublishedStatus%> **From version 6.2.0**	Same as ArticleStatus
<%Rating%>	Outputs the rating on the story. This feature is typically used in movie reviews. The field outputs just rating value and it can be used to show an image like this: `[%{rating>0}<img src="/graphics/<%rating%>.gif">%]`
<%RelatedStories%>	Lists the stories related by taxonomy.

<%Sectioncat%>	The ID for the category in which the story is shown. SectionCat is a flexible way to specify which tem the page should use (without affecting the articles you get from the database).
	• Example 1: You want all teasers in a newslist to use your SPORT article template (sport_art In your NewsItem.pbo, enter:
	`<%Link%>&SectionCat=SPORT`
	The resulting URL in your newslist: `/apps/pbcs.dll/article?AID=/20060922/NEWS/60922080/2162&SectionCat=S`
	Example 2: Get a specific include file, based on the actual *SectionCat parameter:
	`<!--#include virtual="/includes/#sectioncat#_header.inc"-->`
	If **SectionCat** does not exist in the URL, the system will look for {{/includes/default_header.i

<%StorySite%> **From version 6.4.0**	Contains the site code of the story.
<%SourceHostName%>	Outputs the host name (www.source.com) of the site from which the story is taken.
<%SourceName%>	The name of the site from which the story is retrieved. Used with multisite in the tag.
<%StoryCategory%> **From version 6.2.0**	Contains the category ID of the story.
<%StoryDate%> **From version 6.2.0**	Contains the date of the story.
<%StoryID%> **From version 6.2.0**	Contains the lopenr / StoryID of the story.
<%StoryProfileList%>	Lists all profiles in which the current story is listed. This variable will only be available when using the Priority option. Please see MULTIPRIORITY section.
<%StoryRef%> **From version 6.2.0**	Contains the story reference, which differs from <%ImageRef%> when using teasers
<%Summary%>	This is the summary of the story. If the story does not have a summary, the first paragraph will be us To avoid outputting the first paragraph of the story (if a summary does not exist) use `<%ingres%>`. F more information please see `<%ingres%>`.
<%Superheading%>	Outputs the super heading. You can use object scripts to force capitalization of the super heading. F example,<%Superheading$u%> will show as 'THIS IS A TEST'<%Superheading$p%> will show as ' is a test'<%Superheading$n%> will show as 'This Is A Test'<%Superheading$l%> will show as 'this i test'
<%Text%>	Outputs the first paragraph of the story. This field is used with the option **Alltext.**
<%Theme%>	Outputs the theme field for the story.
<%ThemeLinks%>	Outputs stories that have the same theme as the current story. The function uses the object files NewsItemThemeLinks.pbo and NewsItemThemeLinkItem.pbo.
<%Time%>	The time the story was last modified. This field uses the **Timeformat** option in the tag for formatting.
<%Title%>	The story title.
<%TopStory%>	Top story is set on all stories with a priority between 1 and 3. Top story can be used to set class on a as in this example: `<a [%<%TopStory%> class="hot"%] href="<%Link%>"><%Heading%></a> ` `Class="hot"` is only output if the story is a Top story. If the story is not a Top story, no class tag will be output.
<%TotalNoofVotes%> **From version 6.4.0**	Contains the total number of votes on the story.

<%UserSubmitDeleteLink%> **From version 6.2.0**	Contains a direct link that if visited deletes the story (if the user is the story's creator)

NewsItemArticleConnectionItem.pbo

Used with the field <%ArticleConnectionItems%> in NewsItemInternalLinks.pbo.

The file uses the same fields as NewsItemConnectionItem.pbo except <%IS....%> fields are not available.

NewsItemConnectionItem.pbo

Used with the field

```
<%ConnectionItems%>
```

in NewsItemInternalLinks.pbo.

<%\Date%>	The date in localized format.
<%Heading%>	The heading for the object.
<%IsForum%>	This field has the value 1 if the link is to a forum.
<%IsGallery%>	This field has the value 1 if the link is to a photo gallery.
<%ISODate%>	The date in ISO format.
<%Leadin%>	The lead-in for the object.
<%Link%>	The link to the object.

NewsItemExternalLinkItem.pbo

Used with the field

```
<%NewsItemExternalLinkItem%>
```

in NewsItemExternalLinks.pbo.

<%ImageLink%>	Contains the link to the photo defined in Story External Links in the Edit interface.
<%LinkDescription%>	The description of the link.
<%LinkID%>	The ID of the link.
<%LinkMainCategory%>	The main category for the link.
<%LinkRegion%>	The region for the link.
<%LinkSubCategory%>	The subcategory for the link.
<%LinkSubRegion%>	The subregion for the link.
<%LinkTitle%>	The title of the link.
<%LinkURI%>	The link's URL.
<%LogoLink%>	Contains the link to the logo defined in Story External Links in the Edit interface.

NewsItemExternalLinks.pbo

Used to show links to external objects that are linked to the story.

<%NewsItemExternalLinkItem%>	Shows external links.

NewsItemForumConnectionItem.pbo

Used with the field <%ForumConnectionItems%> in NewsItemInternalLinks.pbo.

The file uses the same fields as NewsItemConnectionItem.pbo except <%IS....%> fields are not available.

<%ObjectClass%>	If you need to use the current object class with the forum template, add it to the link like this: `<%Link%>&ObjectClass=<%ObjectClass%>`

NewsItemForumGroupLink.pbo

Used to link directly to a story forum from the Newslist.

<%Link%>	The link to the forum.

NewsItemGalleryConnectionItem.pbo

Used with the field <%GalleryConnectionItems%> in NewsItemInternalLinks.pbo.

The file uses the same fields as NewsItemConnectionItem.pbo except <%IS....%> fields are not available.

NewsItemInternalLinks.pbo

Used to show links to other internal objects that are linked to the story.

<%ArticleConnectionitems%>	Shows links to connected stories.
<%Connectionitems%>	Shows internal links to both stories and photo galleries.
<%ForumConnectionitems%>	Shows links to connected forums.
<%GalleryConnectionitems%>	Shows links to connected galleries.

NewsItemKeywordLinks.pbo

Please see KeywordLinks.pbo for information about available fields.

NewsItemKeywordLinksItem.pbo

Please see KeywordLinkItem.pbo for information about available fields.

NewsItemMultiMediaExtension.pbo

NewsItemMultiMedia[Extension].pbo

Used to link to directly connected multimedia objects. [Extension] is replaced by the media type extension.

***<%Description%> ***	The description of the media object (listed as 'title' in the edit interface).
***<%FileName%> ***	The file name.
***<%FileSize%> ***	The file size.
***<%Link%> ***	The link for direct download of the object.
<%PopupLink%>	The link for the pop-up multimedia window.
***<%StreamLink%> ***	The link for streaming the object.

NewsItemThemeLinkItem.pbo

Please see ThemeLinkItem.pbo for information about available fields.

NewsItemThemeLinks.pbo

Please see ThemeLinks.pbo for information about available fields.

RelStoriesItem.pbo

Used with **RelTaxonomy** when showing related stories based on taxonomy.

<%\Date%>	The publication date for the story.
<%Heading%>	The heading of the linked article.
<%Leadin%>	Outputs the summary if the leadin is larger than 0.
<%Link%>	The address of the connected article.

RelStoriesItems.pbo

Used with **RelTaxonomy** when showing related stories based on taxonomy.

<%ItemCount%>	The number of stories in the list.
<%stories%>	The list of stories.

URL Parameters and overview tags

You can control what the page displays by putting parameters in the URL. This is done by adding **&params=** to the end of the URL.

Calling a Newslist with the URL parameters:

```
&Params=StDato=20050730,SlDato=20060731
```

sets a start date and an end date for the Newslist stories. If nothing is specified, a default range will be used. The default maximum range is 30 days (StDato - SlDato). This can be changed in the ini file. Please see MaxDaysForNewslistDateRange.

StDato	Used to specify the start date. Use YYYYMMDD format.
SlDato	Used to specify the end date. Use YYYYMMDD format.
Slstdato	An alternative way to set the start and end date of the story selection. Possible values could be **SISTE12T**, **SISTE24T**, or **SISTE14D**. If the value given is not one of these three, then it is assumed that the given value is a date in YYYYMMDD format and it will select stories from only that day.

Another option in the URL is to set the profile. When you place a profile in the **Newslist** tag, the profile selected on the **Category** settings page in the edit interface will be used. You can override this profile by entering

```
&Profile=XXX
```

in the URL, where XXX is the profile number. You can specify the profile directly by entering

```
Profile=XXX
```

in the tag. This will lock the tag to display stories matching profile number XXX. You can limit the selection of stories using categories with profiles. This means that stories matching a profile also will have to match the category.

pbs related

<pbs:related>

The theme overview is used for showing all stories with a common theme. Usually, only 5 to 10 links to stories with the same theme are shown when you view the story. The theme overview enables you to show all the stories with the same theme.

This function uses the template **gen_tovr.pbs**. The link to this page is provided by `<%MoreLink%>` in the listing of related stories on the story.

Options

Antall	The number of stories to be shown. By default, the value is all stories will be shown. Enter antall=x where x is any number between 1 and 9999.
Bgcolor	Sets the background color. This color is set in the standard HTML way. This option is ignored when object files are used.
Color	Sets the text color when object files are not used. This color is set in the standard HTML way.
Daterange	By default, only stories from the last 180 days are shown. Enter months=x to show stories from the last x months.

Leadin	This option determines how many characters of the lead-in will be shown. By default, the entire lead-in is shown. Enter
	`leadin=x`
	where x is any value from 0 to 9999. Setting
	`leadin=0`
	turns off the lead-in.

Objectclass	Sets the object class for the tag. By default, the tag will use the ThemeLinks.pbo and ThemeLinkItem.pbo object files. Enter `objectclass=x` where x is any number from 1 to 99 to change the object class.

ThemeLinkItem.pbo and overview tags

<%\Date%>	The story's date. Formatting is controlled by the **Dateformat** option.
<%IsoDate%>	The date in ISO format.
<%Leadin%>	The summary of the story. The amount of text shown is controlled by the **Leadin** option.
<%Link%>	The link to the story.
<%\Title%>	The title of the story.

An example of using these options in the object file might be the following:

```
<li type=square> <a href="<%Link%>"><%\Title%></a> <%\Date%>
```

ThemeLinks.pbo and pbs related

<%Items%>	This field outputs all the items.
<%Theme%>	The name of the theme.

pbs subcategories

<pbs:subcategories>

This tag is used to list subcategories. It can also list the category's stories.

The tag with options can look like this:

```
<table>
<pbs:subcategories force main columns=3 description=0 class=2>
</table>
```

This tag must be placed inside a table if you use multiple columns.

Options

Note that the system will use object files if they exist. All options except class are also available when you are not using object files.

Category	Used to list a specific subcategory.
Class	Determines which object class the tag should use. By default, SubcatSimpleLink.pbo is used. By setting this option to 1, the system will use SubcatSimpleLink1.pbo instead.
Columns	This option controls how many columns the list will have.
Count	Determines the number of stories shown. 0 (zero) outputs all stories.
Force	Makes the system list all subcategories and ignore the **Active** flag in the category register.
Main	Makes the system list all subcategories of the main category being viewed. If this option is not set, only subcategories of the category being viewed will be shown.
Maincategory	Used to list subcategories of the selected main category. The option is set like this: `Maincategory=NEWS`
Showart	Retrieves the stories from the subcategory. By default, this option is off. Change this option setting to On to retrieve stories from the subcategory.
Visref	Determines how many characters of the lead-in will be shown. The option is set like this: `Visref=200`

SubcatSimpleLink.pbo

The object file is output once per category.

<%Article%>	Lists the stories for the subcategory.
<%CategoryName%>	The name of the category.
<%Link%>	The link to the sectionfront for the category.

Chapter 11 Searching

SAXOTECH Online provides the ability to search the stories, forum messages, CCE objects and images in the system. The search functionality has two parts:

- The search form, where the web guest inputs the query.
- The search result page that displays the result.
  Searching CCE objects is also described in the CCE chapter. For more information, please see Search form#FastSearch.

By default, the system uses SQL free text searching in the database. For more efficient searching, Online Search powered by FAST InStream search functionality is implemented. To use the FAST InStream functionality, certain settings must be made in the FASTSEARCH section section of the Publicus.ini file.

- Guided search
  - pbs searchresult_guided
    - ImageSearchNavigator.pbo
    - ImageSearchNavigatorItem.pbo
    - SearchNavigator.pbo
    - SearchNavigatorItem.pbo
    - StorySearchNavigator.pbo
    - StorySearchNavigatorItem.pbo
- Image search
  - pbs imagesearchresult
    - SearchResultImage.pbo
    - SearchResultImageItem.pbo
    - SearchResultImageRow.pbo
- Result page
  - Building the page without object files
    - pbs search_criteria
    - pbssearch_hits
    - pbs search_next_link
    - pbs search_previous_link
    - pbs search_totalhits
    - pbs searchresult without object files
  - pbs searchresult
    - SearchError.pbo
    - SearchResult.pbo
    - SearchResultItem.pbo
- Search form
- Search other websites
- Search syntax

Guided search

SAXOTECH Online Search powered by FAST search functionality has two parts regarding SQL free text searching:

- The search form where the web guest inputs the query.
- The search result page that displays the result.
  All existing search options in the SQL free text search described above can be used. SAXOTECH Online Search powered by FAST search is available in a standard edition as well as an advanced edition. The standard edition has the same options as the SQL free text searching, but with a more efficient SAXOTECH Online Search powered by FAST search engine.

Image Search and Guided Search are only available to users of the advanced edition.

Guided search includes navigators. In order to use navigators, the following option must be included in the search action for stories, forum messages or images:

```
<input type="hidden" name="BuildNavigators" value="1">
```

Below you will find pbo files that must be available when including navigator Guided Search.

pbs searchresult_guided

<pbs:searchresult>

Options

NavigatorList	Specifies which navigators are shown in the result set. If * is used, all navigators will be shown.
SortField	Sorts the result according to the field specified. The default is publication date. A list of possible sort fields is provided in Publicus.ini. Please see SAXOTECH Online Search powered by FAST.
SortDirection	Determines whether the result is sorted in ascending or descending order. The default is descending.

This example shows the result tag for a story and a forum search:

```
<pbs:searchresult Showref=1 Sites="DA" Count=10 navigatorlist="Keywords, Locations" sortfield=
"articlenumber" sortdirection="ASCENDING">
```

This example shows the result tag for image search:

```
<pbs:imagesearchresult Showref=1 Sites="DA" ResultsOnPage=6 navigatorlist="Keywords, Locations"
imagecolumns=3 sortfield="articlenumber" sortdirection="ASCENDING">
```

ImageSearchNavigator.pbo

Used for preparing search in images.

<%NavigatorName%>	The name of the current image navigator.
<%NavigatorHits%>	Specifies the number of hits in the result set used in the image navigator.
<%Modifiers%>	Holds all elements (modifiers) for one image navigator. All modifiers use the ImageSearchNavigatorItem.pbo file.

ImageSearchNavigatorItem.pbo

Used for preparing search in images.

<%ModifierName%>	The name of the current modifier.
<%ModifierHits%>	Returns the number of hits related to the current modifier.
<%ModifierValue%>	Returns a pre-made search string that if chosen should be used as input to search criteria when performing a new search.

SearchNavigator.pbo

Used for preparing search in forum messages.

<%NavigatorName%>	The name of the current forum navigator.
<%NavigatorHits%>	Specifies the number of hits in the result set used in the forum navigator.
<%Modifiers%>	Holds all elements (modifiers) for one forum navigator. All modifiers use the SearchNavigatorItem.pbo file.

SearchNavigatorItem.pbo

Used for preparing search in forum messages.

<%ModifierName%>	The name of the current modifier.
<%ModifierHits%>	The number of hits related to the current modifier.
<%ModifierValue%>	Returns a pre-made search string that, if chosen, will be used as input to search criteria when performing a new search.

StorySearchNavigator.pbo

Used for preparing search in stories.

<%NavigatorName%>	The name of the current story navigator.
<%NavigatorHits%>	Specifies the number of hits in the result set used in the story navigator.
<%Modifiers%>	Holds all elements (modifiers) for one story navigator. All modifiers use the StorySearchNavigatorItem.pbo file.

StorySearchNavigatorItem.pbo

Used for preparing search in stories.

<%Counter%>	Countains the counter value.
<%ModifierName%>	The name of the current modifier.
<%ModifierHits%>	The number of hits related to the current modifier.

<%ModifierValue%>	Returns a pre-made search string that, if chosen, will be used as input to search criteria when performing a new search.

Image search

SAXOTECH Online Search powered by FAST search functionality has two parts regarding SQL free text searching:

- The search form where the web guest inputs the query.
- The search result page that displays the result.

All existing search options in the SQL free text search described above can be used. SAXOTECH Online Search powered by FAST search is available in a standard edition as well as an advanced edition. The standard edition has the same options as the SQL free text searching, but with a more efficient SAXOTECH Online Search powered by FAST search engine.

Image Search and Guided Search are only available to users of the advanced edition.

Image search allows for searching all images related to stories and photo galleries. Image search will search stories in the following text fields found in the SAXOTECH Online Search powered by FAST Index Profile: title, net title, subtitle, byline, body, image texts, image captions.

Template

Image search results are shown using the <pbs:imagesearchresult> tag in the **gen_imagesearch** template. It is also possible to specify a template using the keyword TEMPLATE=TemplateName in the search action.

Image search uses the **imagesearch** action.

Options

StartDate	This is the first date from which the system will select images. This date field must be in the YYYYMMDD format. It can be both visible and hidden.
EndDate	This is the last date from which the system will select images.
BuildNavigator	If BuildNavigator has value="1", navigators defined in the NavigatorList in `<pbs:imagesearchresult>` will be built.
Criteria	The field for entering the search criteria.

SearchCategory	Determines which categories will be searched. You can set the search category in the following ways:
	`<input type="hidden" name="SearchCategory" value="%">`
	`<input type="hidden" name="SearchCategory" value="Categoryid">`
	`<input type="hidden" name="SearchCategory" value="Categoryid1;Categoryid2">`
	`<input type="hidden" name="SearchCategory" value="Categoryid%">`
	`<input type="hidden" name="SearchCategory" value="Categoryid[1-3]">`
	You cannot set the search category using multiple categories and wildcards.

ExcludeCat	The same rules as for setting search categories apply when excluding categories.
SearchProfile	You can limit the search to a profile.
IncludeStatus	Add this field to search stories that are not published.

Example showing the use of the imagesearch action:

```
<form method="post" action="imagesearch?Category=SEARCH">
<input type="hidden" name="EndDate" value="20060606">
<input type="hidden" name="StartDate" value="19600101">
<input type="hidden" name="BuildNavigators" value="1">
<input type="hidden" name="ExcludeCat" value="ARTICLES">
<input type="checkbox" name="IncludeStatus" value="7">
<input type="text" name="Criteria" style="width : 95px" size="8">
<input type=submit name=Go VALUE="Go">
</form>
```

pbs imagesearchresult

<pbs:imagesearchresult>

Options

Navigatorlist	A list of navigators to include in the search result.
ObjectClass	Specifies the object class used.
Sites	Used when you are combining multiple sites in a portal. If you are not searching multiple sites you do not need to add this option. To use this option, enter the site codes in a comma separated list.
ImageColumns	Specifies the number of images shown in one row. The default value is "1".
ResultsOnPage	This option is used for controlling how many results should be shown per page. The default value is "12".
SortField	Sorts the result according to the field specified. The default is the publication date. A list of possible sort fields is provided in Publicus.ini. Please see FASTSEARCH section.
SortDirection	Determines whether the sort is ascending or descending. The default is descending.

Example showing the use of `<pbs:imagesearchresult>`:

```
<pbs:imagesearchresult Showref=1 sites="DA" ResultsOnPage=6 navigatorlist="Keywords, Locations"
imagecolumns=3 objectclass=1 sortfield="news">
```

SearchResultImage.pbo

<%SearchCriteria%>	The text of the search query.
<%SearchPreviouslink%>	The link to previous page of results.
<%SearchNextlink%>	The link to the next page of results.
<%TotalHits%>	The total number of images found.
<%SearchHitRange%>	The value for *from* - *to* hits the page is showing.
<%SearchResult%>	Where the results defined in SearchResultImageItem.pbo and SearchResultImageRow.pbo will be listed.
<%SearchTaxonomy%>	Where the taxonomy tree will be placed.
<%SearchNavigator%>	Where the navigators will be placed.

SearchResultImageItem.pbo

This pbo file is needed to define each result item. The items returned can either be images belonging to stories, or images found in photo albums.

<%DocumentNumber%>	The story number or photo album number depending on the type of result.
<%DocumentByline%>	The byline of stories or photo albums depending on the type of result.
<%DocumentTitle%>	The title of stories or photo albums depending on the type of result.
<%DocumentStatus%>	The status of stories or photo albums depending on the type of result. In photo album's status will always be -9.

<%DocumentDate%>	The publication date of stories or photo albums depending on the type of result.
<%DocumentCategory%>	The category of stories or photo albums depending on the type of result.
<%DocumentLink%>	The document link of stories or photo albums depending on the type of result. For stories, this will link to the relevant story. When an image from a photo album is returned, a link to a photo album will be returned. This requires the appropriate templates.
<%IsFromPhotoAlbum%>	Returns "1" if the image is from a photo album. Otherwise it returns "0".
<%DocumentLeadIn%>	The lead-in from stories.
<%ImageSrc%>	The image result storage location (src), independent of image type (story-connected or photo album).
<%ImageCaption%>	The image caption.
<%ImageText%>	The image text.
<%ImageByline%>	The image byline.

SearchResultImageRow.pbo

Defines the number of images shown in a row. The number of images shown can be specified in `<pbs:imagesearchresult>`.

<%RowItems%>	The number of image results specified in the **ImageColumns** option in `<pbs:imagesearchresult>`. The **ImageColumns** option specifies the number of images that should be shown in one row.

Result page

The result page is used displaying the result page of a search.

Template

The template for displaying the result is called **gen_search.pbs**.

You have two options when building a result page:

- If the object files exist, the page will be built using object files.
- You can use the #tag# tags directly in the search result page.

Building the page without object files

You can place the following tags on the result page.

pbs search_criteria

<pbs:#search_criteria#>

This tag outputs the text of the query.

pbssearch_hits

<pbs:#search_hits#>

This tag outputs the hits range the web guest is viewing.

pbs search_next_link

<pbs:#search_next_link#>

This tag outputs a link to the next page of hits. The tag will only output a link if there is a next page. The label will be shown anyway.

Label	This option sets the text string that should be used for linking to the next hits.

pbs search_previous_link

<pbs:#search_previous_link#>

This tag outputs a link to the previous page of hits. The tag will only output a link if there is a previous page. The label will be shown anyway.

Label	This option sets the text string that should be used for linking to the previous hits.

pbs search_totalhits

<pbs:#search_totalhits#>

This tag outputs the total number of hits.

pbs searchresult without object files

<pbs:searchresult>

This tag outputs the search result.

Infocolor	This color will be used to show the category and date.
Showref	1 = the lead-in of the story will be shown0 = the lead-in of the story will not be shown.

pbs searchresult

<pbs:searchresult>

If you want to use object files, place the `<pbs:searchresult>` tag on the template. If not, you must use the other tags described in this chapter.

The <pbs:searchresult> tag has the following options:

Showref	Determines whether summaries will be retrieved from the database. If **Showref=1** then the summaries will be retrieved from the database.If **Showref=0** then the summaries will not be retrieved from the database. If you do not plan to show summaries, you should use this option to gain speed.
Sites	Used when combining multiple sites in a portal. If you are not searching multiple sites, you do not need this option. To use this option, enter the site codes in a comma separated list.
Alltext	Determines that all the text will be shown for each story, instead of only the summary.
Count	Determines how many results will be shown per page. The default value is 10.

SearchType	SearchType can have two values: Forum or Story. If both stories and forum messages should be searched, two
	`<pbs:searchresult>`
	tags must be placed on the template. One with **Searchtype="forum"** and another with **Searchtype="story"**.By default, if no search type is specified, Search Type will default to **Story**.This is an example taken from the gen_search template:
	`<pbs:searchresult showref=1 navigatorlist="Publishdate,Locations, Names, Companies, Rating" searchtype=story>`
	`<pbs:searchresult showref=1 navigatorlist="Publishdate,Locations, Names, Companies, Rating" searchtype=forum>`
	If a search type is specified in both
	`<pbs:searchresult>`
	and in the /search action, the search type used in
	`<pbs:searchresult>`
	will have precedence over the search action.

RatingSearch	Used to search stories by rating. Example:
	`<pbs:searchresult ratingsearch="1,2,3">`
	will return all stories with a rating of 1, 2, or 3.
	`<pbs:searchresult ratingsearch="2">`
	will only return stories with a rating of 2.It is also possible to add the **ratingsearch** parameter through the standard search form using POST variables or GET variables.

IncludeProfile	Used to include a profile associated to the story category in the result link. This option is only available when using Online Search powered by FAST. Example: `<pbs:searchresult IncludeProfile=1>`
Useobject	Determines whether or not object files are used. The default setting is 0, indicating that object files are not used. To enable object file usage, the option setting should be 1.
Objectclass	Determines which object class is used. By default, the system will use the files in the default object class. Objectclass can be set from 1 to 999.

This tag supports object files.

SearchError.pbo

This object file is output when a search is for less than three characters, or with larger errors like database connection errors.

<%ErrorMessage%>	Contains the error message shown in case of error.

SearchResult.pbo

This is the framework for the search result.

This pages will only be parsed if SearchResultItem.pbo are created.

<%search_criteria%>	The string for which the web guest searched. This tag is similar to the tag you use when you do not have object files.
***<%search_hits%> ***	The value of *from* - *to* hits the page shows. This tag is similar to the tag you use when you do not have object files.
<%search_totalhits%>	The total number of stories found. The system will not return more than 1000 hits. This tag is similar to the tag you use when you do not have object files.

<%search_previouslink%>	The link to the previous page of results. The recommended method to use this criteria is as follows:
	`<form action="search" method="post">`
	`<div id="Prevsearch">Previous`
	`<input type="hidden" name="url" value="<%search_previouslink%>">`
	`<input src="xxxxxxx" alt="search" style="vertical-align: middle;"`
	`type="image"> </div> </form>`

<%search_nextlink%>	The link to the next page of results. It must be placed inside a `<a href="">` tag.The recommended method to use this criteria is as follows:
	`<form action="search" method="post">`
	`<div id="Nextsearch">Next`
	`<input type="hidden" name="url" value="<%search_nextlink%>"> <input`
	`src="xxxxxxx" alt="search" style="vertical-align: middle;" type=`
	`"image"> </div> </form>`

<%Search_result%>	Where the results will be listed.If, for example, no articles are found, then search result returns the message "No Articles Found." This message can be modified using a regular expression such as the following:
	`[%{Search_result="No articles found."}`
	`Your search returned no results. Please modify your search criteria and try again.`
	`%|%`
	`<%Search_result%>%]`

SearchResultItem.pbo

This object file is output for each item the search returns.

<%ArticleAddress%>	The URL of the website on which the story was published. This is only useful for a portal that wants to link to a page on another site.
<%ArticleCategory%>	The category in which the story was published.
<%ArticleDate%>	The story's date field matching the story's <%ISODate%>.
<%ArticleLink%>	The URL for the story.
<%ArticleModifiedDate%> **From 6.2.0**	The story's ModifiedDate field matching the story's <%ISOModifiedDate%>
<%ArticlePublishDate%> **From 6.2.0**	The story's PublishDate field matching the story's <%ISOPublishDate%>
<%ArticleSource%>	The name of the website on which the story was published. This is only useful for a portal that searches multiple sites.
<%Byline%>	The byline for the story.
<%Counter%>	Countains the counter value.
<%LeadIn%>	The lead-in of the story.
<%MainImageUrl%>	Link to main image on story.
<%Rating%>	The story's rating.
<%Status%>	Outputs different messages based on the status code for the story. Please see Article Status for information about the valid statuses. The field can be used like this: `[%{Status=7}<font color="#ff0000"><b>This is a restricted article</b> </font>%]`
<%TeaserUrl%>	The url attached to a teaser
<%Title%>	The title of the story.

Search form

The search form is where the web guest enters her query.

Most field values can also be coded in the **Searchresult** tag or as parameters in the URL, in addition to being set using fields in the search form.

Example form code

```
<form method="get" action="/apps/pbcs.dll/search">
```

To specify the category in which you want the search result displayed, you can add ?Category=categoryname in the action field like this:

```
<form method="get" action="/apps/pbcs.dll/search?Category=SEARCH">
```

By using the line of code below, you can choose which template should be used to display the search results.

```
<input type="hidden" name="Category" Value="CategoryName">
```

For example, without this code, the search result will be shown by the **gen_search.pbs** template.

But with this code:

```
<input type="hidden" name="Category" Value="SPORT">
```

The **SPORT_search.pbs** template is used.

Templates

The search form can be placed on all pages.

Fields

Input field	The field in which web guest enter the search criteria is called Crit. The field is used like this: `<input type="text" name="crit" style="width : 95px" size="8">` This field is used for passing information to the search program.
Start date	This is the first date from which the system will select stories. This date field must be in the YYYYMMDD format. It can be both visible and hidden. You can also use the calendar from the events search for input of this date. You must give the form a name, and change the name of the field in which the date will be entered.
End date	This is the last date from which the system will select stories. Please see **Start date** for more information.

Daterange	This option takes two dates, separated by a comma. This option is often used with the Datecalc function to search the current month. The option can be used like this:
	`<input type="hidden" name="daterange" value="`
	`<pbs:datecalc now firstofmonth days=-1 firstofmonth>,`
	`<pbs:datecalc now firstofmonth days=-1>">`

Include stories without date	Makes the system search stories that are placed on the date 99999999. The option is enabled like this:
	`<input type="hidden" name="IncludeNoDateArt" value="1">`
	or
	`<input type="hidden" name="INDA value="1">`

Search category

Determines the categories that will be searched. This can be either a hidden field or a drop-down menu. When specifying multiple categories, you must use a semicolon as the separator.

- **Setting search category** You can set the search category in the following ways:

```
<input type="hidden" name="SearchCategory" value="%">
<input type="hidden" name="SearchCategory" value="Categoryid">
<input type="hidden" name="SearchCategory"
value="Categoryid1;Categoryid2">
<input type="hidden" name="SearchCategory"
value="Categoryid%">
<input type="hidden" name="SearchCategory"
value="Categoryid[1-3]">
```

  You cannot use multiple categories and wildcards:

```
<input type="hidden" name="SearchCategory"
value="Categoryid%;Categoryid%">
```

- **Drop down example** To create a drop-down menu that lets the web guest select a category, you must build a select box like this:

```
<select name="SearchCategory">
<option value="" SELECTED>All</option>
<option value="NEWS01" SELECTED>Local news</option>
<option value="NEWS02">Regional news</option>
<option value="NEWS03">National news</option>
<option value="NEWS04">World news</option>
</select>
```

Search profile

You can limit the search to a profile. You can build a drop-down for selecting profiles like this:

```
<select name="SearchProfile">
<option value="0" SELECTED>All</option>
<option value="1001">News</option>
<option value="1002">Sports</option>
</select>
```

Specifying a nonexistent profile will result in profiles being ignored. You can also set the profile using a hidden field.

Excluding categories from search

You can exclude specific categories from a search. The option is set like this:

```
<input type="hidden" name="ExcludeCategories" value="Categoryid">
<input type="hidden" name="ExclCat" value="Categoryid1;Categoryid2">
```

The same rules as for setting a search category apply when excluding categories. Please see **Search category**.

Keyword search

This option is used for searching by keyword in the system. It can be used with the Getkeywords function. For more information, please see [pbs GetKeywords.The keyword search is used as in this example:

```
<select name="Keyword" size="1">
<option value=" " SELECTED>Select Topic</option>
<pbs:getkeywords category="NOTICES" options=1>
</select>
```

If you have a lot of keywords, this function will make the page load very slowly.

Restricted stories	You can also search stories that are not published. To do this, add the field **Includestatus.** The field can be used like this: `<input type="checkbox" name="Includestatus" value="7">Search restricted bylines` Please see Article Status for more information about valid statuses

Search other websites

The multisearch function previously described in this section is obsolete. Please use this JavaScript to achieve the same functionality.

This JavaScript allows you to enter sections of your own site to be searched to the select field, in addition to the available search engines. Also, you can add or remove certain search engines, as needed.

This example is taken from a script written for The Royal Gazette. Engine 1 and 9 are local to The Royal Gazette and are handled differently than the search engines.

The URL for the search engines was gathered by performing a search using the search engines and then modifying the search so the script can add the query term to the URL string passed to the individual search engines. Using this method will allow you to add other search engines to this list.

```
<script language=JavaScript>
function EngineCheck() {
var Engine = '';
for (i=0; i<document.form1.engine.options.length; i++) {
if (document.form1.engine[i].selected) {
Engine = document.form1.engine[i].value;
}
}
if (Engine == '1') {
document.form1.action = 'search';
}
if (Engine == '9') {
document.form1.SearchCategory.value = 'MIDOCEAN';
document.form1.action = 'search?Category=MIDOCEAN';
}
if (Engine == '2') {
var Q = document.form1.crit.value;
document.form1.action = 'http://www.google.com/search?sourceid=mozclient&ie=utf-8&oe=utf-8&q=' +
Q;
}
if (Engine == '3') {
var Q = document.form1.crit.value;
document.form1.action = 'http://alltheweb.com/search?cat=web&cs=utf8&q=' + Q +
'&rys=0&_sb_lang=pref';
}
if (Engine == '4') {
var Q = document.form1.crit.value;
document.form1.action = 'http://www.altavista.com/web/results?itag=wrx&q=' + Q + '&kgs=1&kls=0';
}
if (Engine == '5') {
var Q = document.form1.crit.value;
document.form1.action = 'http://search.lycos.com/default.asp?lpv=1&loc=lycoshp&tab=web&query=' +
Q + '&x=0&y=0';
}
if (Engine == '6') {
var Q = document.form1.crit.value;
document.form1.action = 'http://www.hotbot.com/default.asp?query=' + Q +
'&ps=&loc=searchbox&tab=web&provKey=Inktomi&prov=HotBot';
}
if (Engine == '7') {
var Q = document.form1.crit.value;
document.form1.action = 'http://msxml.excite.com/info.xcite/search/web/' + Q;
}
if (Engine == '8') {
var Q = document.form1.crit.value;
document.form1.action = 'http://search.yahoo.com/search?p=' + Q +
'&sm=Yahoo%21+Search&fr=FP-tab-web-t&toggle=1';
}
document.form1.submit();
}
</script>
<form name="form1" method="POST">
<input type="hidden" name="SearchCategory" value="%">
<input type="text" name="crit" size="10" maxlength="100">
<br>
<select name="engine" style="width: 120px;">
<option value="1" SELECTED>Royal Gazette </option>
<option value="9">Mid Ocean News</option>
<option value="2">Google</option>
<option value="3">All the web</option>
<option value="4">Altavista</option>
<option value="5">Lycos</option>
<option value="6">Hotbot</option>
<option value="7">Excite</option>
<option value="8">Yahoo</option>
</select><br>
<input type="button" name="Search" value="  Search &raquo; " onClick="EngineCheck();">
</form>
```

Search syntax

The search syntax is used when the web guest is searching on your pages. This information should be present on your website.

The default search operator is AND.

+, AND

+ is used to determine that the word must appear in the story. This option is not necessary on the first word. The default search operator is AND. This means that all words are required in the story.

Usage

```
michael + jordan + basketball
```

This search will return results where all three words appear.

-, NOT

- is used to select words that must not appear in the story.

Usage

```
michael + jordan - basketball
```

This search will return all stories where *michael* and *jordan* are included, but *basketball* is not included.

%, NEAR

% uses NEAR as the logical operator. This means the words must be near each other in the story.

Usage

```
michael % jordan
```

This search will return all stories where *michael* and *jordan* are close to each other.

" "

This option is used for phrase searches.

Usage

```
"Michael Jordan quits"
```

This search will return all stories where these words appear exactly as written.

OR

Uses OR as the logical operator..

Usage

```
Gothenburg or Sweden
```

This search will return all stories with the word *Gothenburg* or the word *Sweden.* |

( )

This option is used to group searches.

Usage

```
vacation (Holland or Netherlands)
```

This search will return all stories where the words *vacation* and *Holland* or the words *vacation* and *Netherlands* appear.

*****

Used to search for partial words.

Usage The search for

```
foo*
```

will return stories with the words *foot, football* and other words starting with *foo.*

Chapter 12 Classifieds

The Classifieds tags are used to display classified ads.

- Ad inserts
  - Ad inserts using pbs classifieds
  - Ads in stories
  - As a banner ad
  - Importing ad inserts
- Classifieds data retrieval tags
- pbs classifieddetails
  - ClassifiedDetails.pbo
  - ClassifiedPrintDetails.pbo
  - ClassifiedsRelated.pbo
- pbs classifiedsfields
  - ClassifiedFieldsXXX.pbo
  - ClassifiedFieldsXXXItem.pbo
  - ClassifiedFieldsXXXRow.pbo
- Classifieds pbs LastUpdate
- Classifieds search form
- Classifieds setup
- pbs classifieds
  - Classifieds.pbo
  - ClassifiedsGroupHeader.pbo
  - ClassifiedsItem.pbo

Ad inserts

Ad inserts allow you to display a classified ad as if it were a banner ad.

Setup

Setup of ad inserts has three steps.

- You must define the ad insert server in the Publicus.ini. You must add the following line to the GENERAL section

```
AdInsertServer=http://ads.starbanner.com
```

* On the ad server, you must create these folders:
/adinserts
/adinserts/objects

- You must create the file: **config.ini** in the **adinserts** folder. It looks like this:

```
[WHOOP]
Class=70
Category=CAT70017
Days=3
Link=http://www.starbanner.com/apps/pbcs.dll/dclassifieds?Date=<%\Date%>&
Category=
*WHOOP*
&Class=
*70*
&Type=<%Type%>&Lopenr=<%ID%>
```

Category	Used to limit the system to display ads from just one ad category. This option is optional.
Class	Determines which adclass will be shown. This option is mandatory.
Days	Determines how many days the system should look back to find active ads.
Link	Sets the link for the pop-up ad. You must set the category for the object files and the class in the link.

Object files

You must create an object file with the same name as the defined section. To follow the example configuration file, it must be called Whoop.pbo. It can look like this:

```
\[%{Text}<script language="JavaScript1.2" src="http://www.starbanner.com/includes/
functions.js" type="text/javascript"></script>
<link href="http://www.starbanner.com/includes/style.css" rel="stylesheet" type="text/css"
title="Stylesheet">
<a href="Javascript:NewWindow('620', '600', '<%Link%>')")>
<img src="http://imgsrv.starbanner.com/apps/pbcsi.dll/classifiedimg?
Date=<%Date%>&Category=<%Type%>&Lopenr=<%ID%>&Element=1&MaxW=120&Border=0" border=0>
</a>
<b><%Title%></b><br>
<%Text%><br>
%\]
```

The following fields are available:

<%Date%>	The publication date. It can be used for showing the ad's image.
<%Id%>	The adÕs ID. It can be used for showing the ad's image.
<%Link%>	The link to the details page.
<%Title%>	The title of the ad.
<%Text%>	The ad text.
<%Type%>	The category. It can be used for showing the ad's image.

Ad inserts using pbs classifieds

You can use all regular object files for Classifieds with this information. ClassifiedsRelated.pbo has access to all the variables you used in the header of your Advertiser.csv file.

Ads in stories

You must add the following options to the article tag:

```
<pbs:article adinsertspot=GIFTINSERT adinsertwidth=200 adinsertheight=200>
```

Options

Adinsertheight	Sets the height of the ad.
Adinsertspot	Sets the spot for which you want ads. This must match one of the defined spots on the ad insert server.
Adinsertwidth	Sets the width of the ad.

Object file

You now have the following field in the Article.pbo:

<%AdInsert%>	This field outputs the banner ad.

As a banner ad

If you are going to use the ad as a banner ad, you must code the iframe manually as in this example:

```
<iframe frameborder=0 marginwidth=0 marginheight=0 width=140 height=160 scrolling=no
src="http://ads.starbanner.com/apps/pbcsad.dll/adinsert?Spot=
*WHOOPFRONT*
&TARGET=&H=160&W=140">
</iframe>
```

You must change the spot name to match the ads you want to show.

Importing ad inserts

Ad inserts can be imported into SAXOTECH Online using a simple csv file format. This will allow you to use related customer information, similar to the Banner Component, and allow multiple ads that access the same customer information.

InQueue

Your InQueue should be modified by adding a new folder in the same level of the InQueue directory as your stories and Classifieds main folder. It should be named appropriately, based on the name of your ads. Examples are: AdInserts, Wow, or RealEstate. This folder should have the following structure:

```
adinserts/archive
adinserts/discarded
adinserts/pdf
adinserts/pictures
adinserts/logos
```

Multifilter

The Multifilter.ini file must be altered to map the path of the uploaded files and map the fields to the database. Please see the Multifilter documentation for more information on the syntax for setting up the Ad insert information.

Advertiser spreadsheet

You should produce a spreadsheet of your advertisers in the comma delimited format. You should use the fields mapped in your Multifilter.ini to produce the different variables. You can have as many variables as you need for your type of ad. Once these customers are imported, they are not accessible to you in the User Interface, so if there is a mistake, you will need to fix it on the spreadsheet and upload again. Each of the fields in the header line of your file will become a variable, and the name of the column is its variable name.

Example
AdvertisersSummer20040601.csv looks like this:

```
Customer,Name,Address,City,State,Zip,Phone,Email,URL
537795,Basket Barn,272 Route 101,Amherst,NH,03031,(603) 673-2716
700042,Champion Windows,230 Ballardvale St.,Wilmington,MA,,(877) 846-3699
```

In this example, you have 9 variables for each customer. The variable **Name,** has Basket Barn for the first entry, and the variable **Phone** will produce (603) 673-2716. The **URL** variable, although present in the header, has no data for either customer.

Ad manifest spreadsheet

The spreadsheet for the Ad manifest is in the same format as the advertiser spreadsheet. This spreadsheet contains the data for the individual ads, and requires certain information to be present. The file name is compiled from four pieces of information: the customer number, the site code, the ad ticket number and the priority.

Example
Basket Barn's ad for ticket number

```
000002: 537795NS000002[x1]
```

The title and value proposition should be enclosed in double quotes to indicate to SAXOTECH Online that the whole phrase is included as the ad text.

Any appearances of double quotes appearing in the text, such as "Big Jim says, "Buy our cars!"" should be prefaced with a backslash to tell SAXOTECH Online that the text is not finished yet. The text should appear in the Ad manifest like this:

```
"Big Jim says, \"Buy our cars\!\""
```

Example
AdmanifestSummer20040601.csv:

```
Filename,Title, Value Proposition,Category,Start Date,End Date
112589NS000031,"Jiffy Lube","More than an oil change",70320,20040530,20050530
118980NS000014,"Dana Shoes","All Azaleia Sandals, 20% off.",70320,20040530,20050530
```

Images

Ad inserts allow two images: A company logo and the ad itself. The logos can be either GIF or JPG format. The ad can be a JPG, GIF or PDF. These must be uploaded at the same time or after the Advertiser and Ad manifest spreadsheets are uploaded. If a change is made to an image, the Ad manifest must be uploaded again to recognize the change.

Uploading ad inserts

You must upload the spreadsheets and images to the new folders you created in order for SAXOTECH Online to read them into the database.

Standard folders

```
/adinserts/
```

should contain both Advertiser.csv and Admanifest.csv files

```
/adinserts/pdf/
```

should contain any PDF ad files

```
/adinserts/logos/
```

should contain any GIF image files

```
/adinserts/pictures/
```

should contain any JPG image files

The Multifilter will take a little time to accept the files and input them into the database. Once the file has been read, it is moved to the archive folder.

Classifieds data retrieval tags

You can retrieve the values of posted fields using the following tags:

Default date	`<input type="hidden" name="defaultdate" value= "<pbs:!#Defaultdate#>" >`
Descending	`<input type="hidden" name="descending" value= "<pbs:!#Descending#>" >`

From date	`<input type="hidden" name="FromDate" value= "<pbs:!#FromDate#>">`
Range	`<input type="hidden" name="range" value= "<pbs:!#Range#>">`
To date	`<input type="hidden" name="ToDate" value= "<pbs:!#ToDate#>">`
Type	`<input type="hidden" name="type" value= "<pbs:!#Type#>">`

All other hidden fields in the form can be retrieved using the same method.

pbs classifieddetails

<pbs:classifieddetails>

The Classifieds detail page is used to show all details about an ad. This page is typically linked to from an ad listing. The tag is used like this:

```
<pbs:!classifieddetails datefields="Date" dateformat="mmm dd, yyyy">
```

You use the same tag to create the printer-friendly version of the ad. The printer-friendly version is made for users that would like to print out an ad.

Template

The template for Classifieds details is called **gen_classd.pbs**.
The template for the printer-friendly version is **gen_classdp.pbs**. The template should contain very few graphic elements.

Options

Class	Determines which class of classified ad will be shown. The classes are predefined and match the different types of advertising you have (jobs, real estate, cars, and so on).
Datefields	Determines which of the fields in the database contains the date information.
Dateformat	Sets the date format which is to be used when the system displays the ads.
Relatedto	Pulls data from the related table in the database. This is typically customer information. If the field in the database is called CUSTOMER, you can use this option: `Relatedto=CUSTOMER` If you use this option, the object file ClassifiedsRelated.pbo is used to show the related information.

ClassifiedDetails.pbo

All fields in ClassifiedsItem.pbo are available.

Other	You can output the other fields that have been defined in the ini file.
<%RelatedInfo%>	This field pulls in the related information object file.

ClassifiedPrintDetails.pbo

This is used for the printer-friendly version of ads. The same fields as in ClassifiedDetails.pbo are available.

ClassifiedsRelated.pbo

If you use the **Relatedto** option, you can use this field to show the related information. The fields that are available here match the fields you have in the related table.

pbs classifiedsfields

<pbs:classifiedsfields>

This tag is used to populate the search form with relevant data. It extracts information from the database. The tag must use one or more options and be used with the exclamation mark option.

Object files will only be used when the parameter
Optionlist=0 is used. XXX in the object file names is replaced with the value from the field **Field.**

Options

All	This option is used with `Field=Class` to show all categories from all classes. By default, this option is not used. Only categories from the current class are shown. Enter `All=1` for all categories to be shown.
Any	Enables *Any* as a choice in a drop-down menu. This function can only be used when you output an option list. By default, _Any _is not available as an option. Enter `Any=1` to have *Any* as a choice in the drop-down menu.
Class	Sets the ad class. This field must be set, if it is used. Class is set by entering `Class="x"` where x is the ad class.
Columns	Defines how many columns the option list will have. This option can only be used when `Optionlist=0`. The option is set this: `Columns=3`

Dateformat	Sets the date format for date fields. Dateformat is set using the codes found in the section
	`[objectfiles_Object_script#Formatting_Dates|objectfiles_Object_script#Formatting_Dates]`
	For example:
	`Dateformat="mm/dd/yyyy"`

Field	Determines which field should be shown. The option is used like this: `Field=Category` The following options are available: • `Category` - This value will make the system list the categories defined for the ad class. • `Class` - This value will make the system list the ad classes. • `Date` - This value will make the system display the dates for the days in the last week. It will also display options t the last 7, 14 and 30 days. • `Subcategory` - This value will make the system list the subcategories defined for the ad class. • `Week` - This value will make the system display the dates for the days in the last week.
Objectclass	Sets the object class. It is only valid when `*Optionlist=0.*`
Optionlist	Disables the output of an option list, and uses object files instead. By default, an option list is output. Enter `Optionlist=0` to use object files instead.
Related	Specifies the information that should be shown from the related information table. You can then use the related fields with **Field** option.
Vertical	Makes the tag list the options vertically instead of horizontally. The option can only be used with the **Columns** option. The option is used like this: `Vertical=1`

ClassifiedFieldsXXX.pbo

<%items%>	This field outputs the items.

ClassifiedFieldsXXXItem.pbo

This file is output for each item.

<%id%>	The ID for the item.

<%name%>	The name for the item. The fields can be used together like this:
	`<option value="<%id%>"`
	`><%name%></option>`

ClassifiedFieldsXXXRow.pbo

When you use the **Columns** option, you must use the ClassifiedsXXXRow.pbo file.

<%col1%>	This field outputs the first row.
<%col2....9%>	You will have a **col** field for each row.

Classifieds pbs LastUpdate

<pbs:!#LastUpdate#>

Outputs the update time for the category being viewed.

Classifieds search form

The search form enables searching of the classified ads. The search form can be put on all pages.

Template

The result uses **gen_class.pbs**. This page must have the pbs classifieds tag to show the results.

Form and fields

The tag pbs classifiedsfields is used in some of the fields to extract information from the database.

Form code

```
<form name="classfsearch" method="POST" action="<pbs:prog>/classifieds?
Category=<pbs:categoryid>">
```

The form code and hidden fields in the example code must be present. If this code is missing, the search form will not work.

Fields

Almost all options that you can use in the tag can also be used as a field in the form. They are either set using hidden fields, or as drop-down in which the web guest can select a value. The fields are documented in the section pbs classifieds

Free text search

You can enable free text searching of ads with the following fields:

```
<input type="text" name="TextSearch" value="<pbs:!#TextSearch#>" size="20">
```

The tag

```
<pbs:!#TextSearch#>
```

is used to output the text the web guest entered on the result page. This is necessary for the next/previous browse function. This is a generic function, and by replacing text with the name of another field, you can search that field instead.

Examples

Below are some examples on how you can search the ads.

From/to search
You can search from and to values. For this to work, you must define two fields:

From	`<input type="text" name= "fromDocketNumberSearch" value= "<pbs:!#fromDocketNumberSearch#>" size="8">`
To	`<input type="text" name= "toDocketNumberSearch" value= "<pbs:!#toDocketNumberSearch#>" size="8">`

This enables search on any value between the from and to values. This can also be used with dates, so you can make a form for searching date ranges. When this function is used, you must remove `Single=1` from the pbs classifieds tag.

Selecting ad category

Used to select the ad category. Use the pbs classifiedsfields to list all categories for the ad class.

```
<select name="Type" style="width: 330px;">
<pbs:!classifiedsfields field="Category" optionlist=1 any=1 class="0">
</select>
```

Classifieds setup

Before you can start using the Classifieds Module, you must set it up.

Publicus.ini file

In SAXOTECH, Online there are three possible levels of classification in the Classifieds engine. At the top are ad classes, which are broad groupings of records that share similar rules. Below classes, you may optionally create groups of categories. Categories are the lowest level classification for records.

In the Publicus.ini, make the following settings:

ADCLASSES section
ADCLASSES section
CacheClassifieds

Categories

Category values correspond to your Classified system's categories. Your categories must be imported into the system by a database administrator.

Customers

Customer meta data (for example advertiser's address, phone number, e-mail, or other static meta data) related to classified ads, can be imported through a .csv file. This is a special import for display (WOW) ads, and is not intended to be used with regular classified ads. The information can be shown using the **relatedinfo** option.

Customer meta data **must** be imported into the same ad class as the ads.

pbs classifieds

<pbs:classifieds>

The main listing is used for giving an overview of ads in the system. This template usually has a search form enabling web guests to find items.

Template

The default template for displaying the list of ads is called **gen_class.pbs**. The template for displaying the ad details is called

gen_classd.pbs. The template for displaying the print version is called **gen_classdp.pbs**.

Options

If you have a search form for the classified ads on the same page as the listing, the options set in the form will override the options in the tag. You can also put parameters in the URL. These will override the form.

Cached	Used to make settings for caching. By default, this option is not used and caching is enabled. If you are using cache qualifiers and cache time in a macro for use with a Classifieds tag and the only difference in the data being fed from the macro into the Classifieds tag is the date, then `Cached=0` must be used in the Classifieds tag to ensure that the cache layers do not conflict. To ensure cache layers will be correct when loading multiple Classifieds tags within macros on the same template, `Cached=0` should be used within the Classifieds tag.
Class	Determines which class of Classified ads the tags will show. The classes are predefined and match the different types of advertising you have, such as jobs, real estate or cars. Please see Classifieds setup.
Count	Specifies the number of ads per page.
Datefields	Determines which of the fields should be treated as dates. This option causes the data to be formatted according to the **Dateformat** options.
Dateformat	Sets the date format for the **Datefields** parameter. **Datefields** can also be formatted in the object files.

Daterange	Sets both the start and end date with the same option. The dates are specified in ISO format:
	`Daterange=20030715-20030722`
	You can set **FromDate** and **ToDate** to the same date by specifying only one date in the **daterange** field like this:
	`Daterange=20030427`

Days	This option has two uses: • It can be used with the random option to determine from how many days the system should show random ads. • It can be used to specify how many days back the system should look for ads. For the second use, you must format it like this: `Days=7 FromDate=-1 ToDate=YYYYMMDD` **FromDate** must be set to -1. **ToDate** specifies the end date for the list.
Defaultdate	This option sets **FromDate** to the last published date. The option is set like this: `Defaultdate=1.`
Descending	Sets the reverse sort order. The content is sorted by the sort order field. The field is used like this: `Descending=1` Setting Descending=0 gives you the normal sort order.

Execute	Deactivates the initial search. By default, the search is executed.If you enter `Execute=0` in the tag, you must set execute to 1 in the search form using a hidden field to see any search results. Otherwise the search will not be executed.

FIELDMultiSearch	Used to search a specified field for multiple keywords. Replace
	`FIELD`
	with the name of the field. The option is used like this:
	`FIELDMultiSearch="text1,text2"`

FIELDMultiSearchAnd	Determines that multiple keyword searches in a specified field will use the logical operator AND. The default is to use the logical operator OR.Replace
	`FIELD`
	with the name of the field. The option is used like this:
	`FIELDMultiSearchAnd=1.`

Fields	Determines which extra fields should be available. All fields defined for the Adclass can be used. The option is set like this: `fields= "County,CaseType,DocketNumber,DateOfVerdict,Judge,Pattorney,Dattorney"` The **Fields** option should be used before any Groups, SortOrder or Search options in the tag. Please note that if you group by a Field, you should not use that field in the **SortOrder** option.

FIELDSearch	Used to search a specified field. Replace FIELD with the name of the field. The option is used like this:
	`FIELDSearch="text"`
	For example:
	`<pbs:classifieds module=4 class=41 fields="Condition" ConditionSearch= "sunny">`
	The parameter **FIELDSearchExact=1** must be used in order for FIELDSearch to work.

FIELDSearchExact	Disables wildcard searches. The text in the criteria must be an exact match. The option is used like this: `FIELDSearchExact=1.`
FromDate	Sets the start date for the list. The date is specified in ISO format: `FromDate=20030715`

FromFIELDSearch	Used to search a field for a specific value. By default this option is not used.To use this option, you must specify `FromFIELDSearch="value".` You must also specify the **ToFIELDSearch** value.
Groups	Sets the group header. This is not to be confused with group mappings in the ini file. If you use the County field to define a group, the file ClassifiedsGroupHeader.pbo will be output when the county changes. The option is set like this: `Groups="County"` The field specified in the **Groups** option will be available as a field in the object file. Groups should be used after the **Fields** option in the tag.
Intfields	Determines which fields should be treated as numbers.
Locked	This option can only be used in the tag. It locks the tag so it cannot be overridden either in the form or the URL. This option is useful when you have a sidebar with ads, which you do not want to be changed by other Classified tags on the page. The option is set like this: `Locked=1.`
Objectclass	This option sets the object class for the object files.
Published	This field is used with the **Range** field to control which ads are shown. It is used like this: `Published=1.` When you use this option, only ads that are still active on the current date will be shown. If you do not use this option, the current status is ignored.
Random	Makes the tag show a random sampling of classified ads. The option is set like this: `Random=1.` By default, the tag looks back seven days. You can override this using the **Days** option.

Range	When you use the option `Range=1` the system will look for ads that were active within the *Daterange.
	• **Rule 1:** Fromdate and Todate are the same and
	`Range=1`
	All ads that were active on the specified date are shown.
	• **Rule 2:** Fromdate and Todate are different and
	`Range=1`
	All ads that were active within the specified interval are shown.
	• **Rule 3**. Fromdate and Todate are the same and
	`Range=0`
	All ads that were published on the specified date are shown.
	• **Rule 4**. Fromdate and Todate are different and
	`Range=0`
	All ads that were published within the specified interval are shown.

RelatedFieldName	Determines which value from the related information table will be shown. By default, the option is not used. Enter `RelatedFieldName="NAME"` to make the value of the name field available in the ad. Only one field name can be defined.
RelatedTo	Enables you to use a field to look up another field from the related information table. The field you use must be defined in the **Fields** option. By default the option is not used. Enter `RelatedTo="Customer"` in the tag. Information from the customer field on the ad will be used to look up information from the related information table.
Single	Determines that a single date is sent to the system. This option operates in conjunction with the search form. Please see Classifieds search form. If this option is not set, you must have **FromDate** and **ToDate** fields.By setting `Single=1` you can supply one date, and the system will show the ads for that date.
SortOrder	Sets the sort order of the data. You can specify all available fields here such as: `Sortorder="County,CaseType,DocketNumber"` SortOrder should be used after the **Fields** option. Please note that if you group by a Field, you should not use that field in the SortOrder option as well.
Start	Sets the start record. This option is typically not used manually. It is used automatically by the system when you have multiple pages of ads.

ToDate	Sets the end date for the list. The date is specified in ISO format: `ToDate=20030715`
ToFIELDSearch	Used to search a field for a specific value. By default this option is disabled. To use this option, you must specify `ToFIELDSearch="value".` You must also specify the **FromFIELDSearch** value.
Type	Used to specify the ad category. This option is only used when you are listing one category: `Type="CAT2900"`
Types	This option works like **Type,** but you can specify multiple ad categories. They must be specified in a comma separated list: `Types="CAT2900,CAT2950"`

Reset drop-down field values

You can use a JavaScript to reset the drop-down fields to the same values that they had before the form was submitted. The following JavaScript sets the value for the fields **ToDate, FromDate** and **Type:**

> ⚠ JavaScript is case sensitive, so you must write the field names exactly the same way as they are in the form. This script must be placed at the bottom of the gen_class.pbs template.

```
<script language="javascript">
var a = document.classfsearch.ToDate.options;
for (x=0;x<a.length;x++) {
if (("<pbs:!#ToDate#>" == "") && (a[x].value == "-1")) {a[x].selected = true;} else
if (a[x].value == "<pbs:!#ToDate#>") {a[x].selected = true;}
}
a = document.classfsearch.FromDate.options;
for (x=0;x<a.length;x++) {
if (("<pbs:!#FromDate#>" == "") && (a[x].value == "-1")) {a[x].selected = true;} else
if (a[x].value == "<pbs:!#FromDate#>") {a[x].selected = true;}
}
a = document.classfsearch.Type.options;
for (x=0;x<a.length;x++) {
if (a[x].value == "<pbs:!#Type#>") {a[x].selected = true;}
}
</script>
```

Classifieds.pbo

This file contains the framework, which is output once. This object file must be present.

<%FirstHit%>	The number for the first record being viewed.
<%Items%>	Outputs the Classifieds items.
<%LastHit%>	The number for the last record being viewed.

<%NextLink%>	Contains a link to the next page, if you are not on the last page. It must be posted to the server with JavaScript like this: `<a href="javascript:dosubmit('<%NextLink%>');">Next »</a>`

<%PrevLink%>	Contains a link to the previous page, if you are not on the first page. It must be posted to the server with JavaScript like this:
	`<a href="javascript:dosubmit('<%PrevLink%>');">`
	`« Previous</a>`

<%SortBy....%>	You can use other fields for sorting , in addition to date. Replace "..." with the name of the field.

<%SortByDate%>	Contains a link that will sort all the ads by date. It must be posted to the server with JavaScript like this:
	`<a href="javascript:dosubmit('<%SortByDate%>');">\\`
	` Published</a>`
	The system will remember the sort order so you can reverse the list. To keep the sort order when you browse using a search form, you must add a field to the form that remembers the value.

<%TotalHits%>	The total number of records for the category being viewed.

Javascript

The following script is used in this file:

```
<script language="javascript">
var hasclicked = 0;
function _Check() {
if (hasclicked == 1) {return false;}
}
function dosubmit(t) {
var f = document.classfsearch;
if (t != "") {f.action = t;}
f.submit();
hasclicked = 1;
}
</script>
```

This blocks the user from submitting the form multiple times. This script requires that the name of your form is "classfsearch". When you use this code in conjunction with the **Next** and **Previous** links, the search form on the page is submitted. To enable this, you must have a search form on the results page and all the fields that you used for your first search must be present and loaded with their original values.

If you have multiple search forms on the same page, they must have different names. Otherwise the script will not work.

ClassifiedsGroupHeader.pbo

Contains one field with the same name as the field specified by the **Groups** option. If you use the option

```
Groups="County"
```

you will have the field

```
<%county%>
```

in the object file. If you have specified a date field in the **Groups** option, the field will be formatted as a date, if it is also listed in the **datefields** option.

This object file must be present.

ClassifiedsItem.pbo

This file is output for each classified ad. Fields that have been configured in the ini file for the ad class are also available. This object file must be present.

<%Category%>	This field contains the category ID for the ad.
<%Categoryname%>	This field contains the category name for the ad.
<%Count%>	This field contains the counter for the item.
<%\Date%>	The date for the classified ad.
<%Dlink%>	Contains a link to the detail page for the ad.
<%Even%>	This field contains 1 if the counter for the item is even.
<%Expires%>	The date the ad expires.
<%ImageLink%>	Contains a link to the image, if the ad is a display ad. It must be used like this: `<img src="<%ImageLink%>&MaxW=320" border=0>`
<%LogoLink%>	Contains a link to the logo, if the ad is a display ad. It must be used like this: `<img src="<%LogoLink%>&MaxW=320" border=0>`
<%Odd%>	This field contains 1 if the counter for the item is odd.
<%PrintLink%>	Contains a link to the printer friendly page. It must be used like this: `<a href="<%PrintLink%>">Print</a>`
<%Relatedinfo%>	This field outputs the information from the related field when you use the **Relatedto** function.
<%Text%>	This field outputs the defined field text, which is defined in the ini file.
<%TipLink%>	Contains a link to the Tip a Friend page. It must be used like this: `<a href="<%TipLink%>">Tip a friend</a>`
<%\Title%>	The title field for the ad.

Chapter 13 Custom Content Engine

The Custom Content Engine (CCE) enables you to create, manage and present your own custom objects. Such objects are typically structured information, such as weather, stock quotes, sports results, marketplace/classifieds, recipes, reviews, etcetera. It is especially suited to large volumes of structured data as it is your own custom database. CCE Objects may also serve as classifieds content for OAMS.

CCE tags are used to display the CCE objects.

- CCE data retrieval tags
- CCE pbs LastUpdate
- CCE search form
- CCE setup
- CCE tip a friend
- pbs cce
    - CCE_ModuleShortName.pbo

- CCE_ModuleShortNameGroupHeader.pbo
- CCE_ModuleShortNameItem.pbo
- cce_ModuleShortNameNavigator.pbo
- cce_ModuleShortNameNavigatorItem.pbo
- CCE_ModuleShortNamePageLink
- CCE_ModuleShortNamePageLinkItem
- CCE_MPLACE.pbo
- CCE_MPLACEitem.pbo
- CCE-search_ovr.pbs

- pbs ccedetails
  - CCE_ModuleShortNameDetails.pbo
  - CCE_ModuleShortNamePrintDetail.pbo
  - CCE_ModuleShortNameRelated.pbo
- pbs ccefields
  - CCE_ModuleShortNameFieldsFdNmRw
  - CCE_ModuleShortNameFieldsFieldName.pbo
  - CCE_ModuleShortNameFieldsFieldNameItem.pbo
- pbs usersubmitform
- SubmitCCEObject action

CCE data retrieval tags

You can retrieve the values of posted fields using the following tags. All other hidden fields in the form can be retrieved using the same method.

Type	`<input type="hidden" name="type" value= "<pbs:!#Type#>">`
Range	`<input type="hidden" name="range" value= "<pbs:!#Range#>">`
Descending	`<input type="hidden" name="descending" value= "<pbs:!#Descending#>" >`
Defaultdate	`<input type="hidden" name="defaultdate" value= "<pbs:!#Defaultdate#>" >`
FromDate	`<input type="hidden" name="FromDate" value= "<pbs:!#FromDate#>">`
ToDate	`<input type="hidden" name="ToDate" value= "<pbs:!#ToDate#>">`

CCE pbs LastUpdate

<pbs:!#LastUpdate#>

Outputs the update time for the category being viewed.

CCE search form

The search form allows web guests to search the CCE Objects.

Template

The search form can be placed on all pages. The result uses **gen_cce_ModuleShortName.pbs**. This page must have the <pbs:cce> tag to show the results.

Form

The tag <pbs:ccefields> is used in some of the fields to extract information from the database.

```
<form name="ccefsearch" method="POST" action=
"<pbs:prog>/cce?Module=Module&Category=<pbs:categoryid>">
```

The form code and hidden fields in the example code must be present. If this code is missing, the search form will not work.

Also, you must add the variable **<%SearchString%>** to the submit form for the link to the next page to return the same search string as the original search. **<%SearchString%>** contains the actual search string.

Fields

Almost all options that you can use in the tag can also be used as a field in the form. They are either set using hidden fields, or as a field where the web guest can select the value. The fields are documented with the <pbs:cce> tag.

In order to search custom fields, you cannot just use the name of the field. You must append "search" to the end of the custom field name, as in the following example for a custom field called "cost":

```
<input type="text" name="costsearch" />
```

As you can see from the example above, in order for the form fields to work, the names of the fields must be in the format /field namesearch.

Free text search

You can also use free text search on the CCE objects. Use the following fields:

```
<input type="text" name="TextSearch" value="<pbs:!#TextSearch#>" size="20">
```

This field is used for free text searches. The tag **<pbs:!#TextSearch#>** is used for outputting the user-entered text on the result page. By replacing text with the name of another field, you can search that field instead.

It is important to note that Full Text search using FAST Search does not search through custom-fields. In order to (also) search the custom fields you must write a search expression using direct acces to an XML field in the FAST Schema containing all contents including contents of custom-fields.

```
<pbs:cce module=4 class=41 search="xml#ford" sortfield="pubdate">
```

The above example wil find all cce objects where the word "ford" is in there somewhere. You can also use this xml-field to search in specific fields as in the following example:

```
<pbs:cce module=4 class=41 search="xml#brand#ford" sortfield="pubdate">
```

Here it must be the field "brand" that is equal to "ford" in order for a cce object to be found.

In order to allow the reader to type in a search criteria that also is used to search custom fields some java-script trickery is needed to transform what the user type to an expression that makes FAST search there.

More examples

Below are some examples on how you can search the CCE objects.

From/to search
You can also search from and to values. For this to work, you must define two fields:

From:

```
<input type="text" name="fromDocketNumberSearch"
value="<pbs:!#fromDocketNumberSearch#>" size="8">
```

To:

```
<input type="text" name="toDocketNumberSearch"
value="<pbs:!#toDocketNumberSearch#>" size="8">
```

This will allow you to search any value between the from and to values. This can also be used with dates, so you can make a form for searching from/to date. When this function is used, you must remove `Single=1` from the `<pbs:cce>` tag.

Selecting type

```
<select name="Type" style="width: 330px;">
<pbs:!ccefields field="Type" optionlist=1 any=1 class="0">
</select>
```

Used to select a type. Use the ccefields tag to list all categories for the class.

Online Search powered by FAST in CCE objects

Online Search in CCE can use *near, and, or* and *not* operators to build complex search expressions.

The search parameter is used to set the search expression. See Search syntax.

The SortField parameter is used to set to the sort field. This search can be used in combination with the Module, Class, Range, Fromdate, Todate and Locked parameters.

Text search examples	`<pbs:cce module=4 class=41 search="mobilehome" sortfield="title">`
	`<pbs:cce module=4 class=41 search="title#mobilehome" sortfield="pubdate">`
	`<pbs:cce module=4 class=41 search="xml#brand#ford"`
	`<pbs:cce module=4 class=41 search="xml#brand#ford and mustang">`
	`<pbs:cce module=4 class=41 search="musta*">`
	`<pbs:cce module=4 class=41 search="xml#brand#ford or xml#brand#toyota">`
	`<pbs:cce module=4 class=41 search="xml#brand#ford and not xml#state#CA">`
	`<pbs:cce module=4 class=41 search="ford near mustang">`
	`<pbs:cce module=4 class=41 range=1 fromdate=20060101 todate=20061230 locked=1 search="ford">`

Value search examples	`<pbs:cce module=4 class=41 search="xml#wind#'2'">` `<pbs:cce module=4 class=41 search="xml#brand#'range(2,10)'">`

CCE setup

Publicus.ini settings

To activate the CCE menu in the user interface, you must add the following value to the **MODULES section** of the Publicus.ini:

```
[MODULES]
CCE=1
```

In the CCE section you can add the following values:

```
[CCE]
CCEClassifieds=0
ClassifiedsModule=11
Cache=1
NoMDCache=1
```

CCE components, classes and types

The first thing to do when creating a new solution with CCE is to use the user interface to setup the meta data for the solution.

- Create the component that will be used when looking up template and object files.
- Create classes for the component.
- For each class, you should specify the required custom fields.
- Add types, subtypes and detail types for the component and/or the classes.

Reserved field names

The following names are reserved and cannot be used as extra custom field names:

- Title
- Date
- Text
- Keyword
- Keywords
- Site
- Category
- Type
- ID
- Expires
- CategoryName
- TypeName
- Image
- AttachedLink
- NavigationCategoryName
- NumImages
- SubType
- SubTypeName
- DetailType
- DetailTypeName

Other setup

Meta data (for example advertiser's address, phone number, e-mail, or other static meta data) related to CCE objects can be setup through the user interface. The information can be shown using the **Relatedinfo** option.

CCE tip a friend

This uses CCE templates for the message. Sending a message uses the file **CCE_<ModuleShortName>TipsText.pbo** or *CCE_<ModuleShortName>TipsTextHtml.pbo. *

The CCE send tip form will use **CCE_<ModuleShortName>tip.pbs** if it is present. Otherwise, it will use the **gen_tip.pbs** template.

pbs cce

<pbs:cce>

The main listing is used to give the overview of CCE objects for the given module For example:

```
<pbs:cce module=4 class=41 fields="Temperature,Condition">
```

The template usually has a search form, enabling web guests to find information.

Templates

The module's short name is used when looking up the main listing template.

The default template to display the list of CCE objects for a module is called **gen_cce_ModuleShortName.pbs**

The template for displaying the object details is called **gen_cced_ModuleShortName.pbs**

The template for displaying the print version is called **gen_ccedp_ModuleShortName.pbs**

Example
If you have a *Weather Data* module, with the short name *weather,* the following templates are used:

gen_cce_weather.pbs
gen_cced_weather.pbs
gen_ccedp_weather.pbs

Actions

The /cce and /dcce CCE URL actions will show the main CCE listing of CCE objects, or will show other templates if you use other options.

Action options

Category	Use the given category as prefix instead of the standard general template.
Module	Resolves the template name.
Template	Use the given template as part of the template name of standard template.

Example
The request

```
/cce?module=4&template=winterweathersearch
```

uses a template file named gen_winterweathersearch.pbs

Options

If you have a search form for CCE objects on the same page as the listing, the options set in the form will override the options in the tag (unless you use the locked option). You can also put parameters in the URL, which will override the form.

Cached	Disables the cache in special circumstances. By default, this option is not used.
	`Cached=0`
	turns off the caching. This is not recommended. Example:
	`<pbs:cce module=4 class=41 cached=0>`

Class	Determines the class of CCE objects that the tags will show. The classes are predefined and match the different types of objects you have in a component. You can setup the classes in the user interface. Example: There is a Weather Data module with ID 4 and the short name *weather*, and there is a Class 41, which belongs to module 4:
	`<pbs:cce module=4 class=41>`
	SAXOTECH recommends using the class parameter. When you use the <pbs:cce> tag WITHOUT a **class=** parameter, you get a list of objects returned whose class=-1 in the <%dlink%> field of the cce_moduleshortnameitem.pbo file. This will NOT populate the dcce template with the objects field information from the database. In order to build a legitimate dlink from a list of CCE objects for which class is NOT set in the tag, perform a regex operation similar to the following:
	`<a href="<%dlink$r("(&Class\x3D-1)=")%>&class=<%class%>">click here</a>`

Component	Sets the component used by the tag. The module's short name is used to look up the templates. This option is required. Example: There is a Weather Data module with ID 4 and the short name *weather*. `<pbs:cce module=4>`
Count	Specifies the number of objects per page. Example: `<pbs:cce module=4 class=41 count=15>`
CreatedBy **From version 6.2.0**	Enables filtering on creator. **#CurrentUser#** is available for showing the current user's submitted objects.
Datefields	Determines which of the fields should be treated as dates. This option causes the data to be formatted according to the **Dateformat** options. Example: `<pbs:cce module=5 class=51 fields="Birthday" DateFields="Birthday">`
Dateformat	Sets the date format for the date fields. The date fields can also be formatted in the object files.

DateRange	Sets both start and end date with the same option. The dates are specified in ISO format:
	`Daterange=20030715-20030722`
	You can set fromdate and todate to the same date by specifying only one date in the daterange field like this:
	`DateRange=20030427`
	Example:
	`<pbs:cce module=4 class=41 daterange="20051101-20051201">`

Days	This option has two uses. It can be used with the **Random** option to determine from how many days the system should show random CCE objects. The second use is to specify how many days back the system should look for CCE objects. When you use the option this way, you must format it like this:
	`Days=7 Fromdate=-1 Todate=YYYMMDD`
	Fromdate must be set to -1. **Todate** specifies the end date for the list. Example:
	`<pbs:cce module=4 class=41 days=14>`

Defaultdate	Sets **Fromdate** to the last published date. The option is set like this:
	`Defaultdate=1`
	Example:
	`<pbs:cce module=4 class=41 defaultdate=1 todate="20051201">`

Descending	This option is obsolete. Use **SortOrder** instead (see below). Enter
	`Descending=0`
	to use the normal sort order. Enter
	`Descending=1`
	to set the reverse sort order. Example:
	`<pbs:cce module=4 class=41 fields="Temperature" Sortorder="Temperature" Descending=1>`

Execute	Deactivates the initial search. By default, the search is executed. If you enter
	`Execute=0`
	you must set execute to 1 in the search form using a hidden field. Otherwise you will not see any search results. Example:
	`<pbs:cce module=4 class=41 execute=0>`

Fields	Determines which extra custom fields will be available. All fields in the field definitions can be used. The option is set like this: `<pbs:cce module=4 class=41 fields="Temperature,Condition,Wind">`

FIELDMultiSearch	Searches a specified field for multiple keywords. Replace FIELD with the name of the field. The option is used like this:
	`FIELDMultiSearch="text1,text2"`
	Example:
	`<pbs:cce module=4 class=41 fields="Town" TownMultiSearch="Boston,Houston" >`

FIELDMultiSearchAnd	Determines that the multiple keyword search in a specified field use AND as the logical operator. The default is to use OR. Replace FIELD with the name of the field. The option is used like this:
	`FIELDMultiSearchAnd=1`
	Example:
	`<pbs:cce module=4 class=41 fields="Town" TownMultiSearch="Las,Vegas" TownMultiSearchAnd=1>`

FIELDSearch	Searches a specified field. Replace FIELD with the name of the field. The option is used like this:
	`FIELDSearch="text"`
	Example:
	`<pbs:cce module=4 class=41 fields="Condition" ConditionSearch="Sunny">`

FIELDSearchExact	Turns off wildcard searches. The text in the criteria must be an exact match. The option is used like this:
	`FIELDSearchExact=1`
	Example:
	`<pbs:cce module=4 class=41 fields="Condition" ConditionSearch="Sunny" ConditionSearchExact=1>`

FromDate	Sets the start date for the list. The date is specified in ISO format: `FromDate=20030715`

Groups	Sets the group header (not to be confused with group mappings in the ini file). The groups option can only be used on custom fields. For example, if you determine that the field *County* is the group header, the file **ModuleShortNameGroupHeader.pbo** will be output whenever the county changes. The option is set like this:
	`Groups="County"`
	The field specified in the groups option will be available as a field in the object file. Example:
	`<pbs:cce module=4 class=41 fields="Temperature" Groups="Country">`
	Another example:
	`Groups="CategoryName"`
	and the pbo file:
	`<!-- CCE_townGroupHeader.pbo-->`
	`<h2><%CategoryName%></h2>`

Intfields	Determines which fields should be treated as numbers. This option is deprecated. Use a custom field with the type *Integer* instead.
Locked	This option can only be used in the tag. It locks the tag so it cannot be overridden, either in the form or the URL. This option is useful when you have a sidebar with objects that should not be affected by other CCE tags on the page. The option is set like this: `Locked=1.`
Multisite	If you cooperate with other media sites, you can use this option to list CCE objects from several sites. Example: There is a Weather Data module with ID 4 and the short name *weather*. `<pbs:cce multisite="BA,TY,FC" module=4>`

NavigatorList **From version 6.4.0**	It is now possible to add navigators to a solr search result. This is done by adding the fields to build navigators for, to the NavigatorList parameter of the CCE tag that should output the result. The navigators are built using the two templates cce_ModuleShortNameNavigator.pbo and cce_ModuleShortNameNavigatorItem.pbo. `<pbs:cce navigatorlist="module,class,type,date,CustomField1" module=4>`
Objectclass	Sets the object class for the object files. The object class is used to look up the object template files for the tag. Example: `<pbs:cce objectclass=49 module=4>` will use object file CCE_Wweather49.pbo if it exists. Otherwise, it will use object file **Weather.pbo**
Profile	Looks up the profile with the given ID and uses the taxonomy expression for that profile to create a query that returns the CCE objects matching the taxonomy expression. If you have a story about a subject that belongs to a profile, you can also show CCE objects that match that profile. Example: `<pbs:!cce module=4 class=41 profile=1334>`

Published	Used with the **Range** field to control which objects are shown. The option is set like this:
	`Published=1`
	When you use this option, only objects that are still active on the current date will be shown. If you do not use this option, the current status is ignored. Example:
	`<pbs:cce module=4 class=41 daterange="20051101-20051201" published=1>`

Random	Makes the tag show a random sampling of objects. The option is set like this:
	`Random=1`
	By default, the tag looks back seven days. You can override this using the **Days** option. Example:
	`<pbs:cce module=4 class=41 random=1>`

Range	If you enter the option `Range=1` the system will look for objects that were active within the Daterange. Example:
	`<pbs:cce module=4 class=41 daterange="20051101-20051201" range=1>`
	* **Rule 1:** Fromdate and Todate are the same and
	`Range=1`
	All objects that were active on the specified date are shown.
	• **Rule 2:** Fromdate and Todate are different and
	`Range=1`
	All objects that were active within the specified interval are shown.
	• *Rule 3: *Fromdate and Todate are the same and
	`Range=0`
	All objects that were published on the specified date are shown.
	• **Rule 4:** Fromdate and Todate are different and
	`Range=0`
	All objects that were published within the specified interval are shown.

RelatedFieldName	Defines which value from the related information table will be shown. By default, this option is not used. Enter
	`RelatedFieldName="NAME"`
	to make the value of the name field available in the ad. Only one field name can be defined. Example:
	`<pbs:cce module=4 class=41 fields="Temperature,Town" RelatedTo="Town" RelatedFieldname="Name">`

Related	Uses a field to look up another field from the related information table. The field you use must be defined in the **Fields** option. By default, this option is not used.Example: If you enter
	`Related="Town"`
	in the tag, information from the town field on the ad will be used to look up information from the related information table.
	`<pbs:cce module=4 class=41 fields="Temperature,Town" Related="Town" RelatedFieldname="Name">`

SearchTaxonomy

Uses the taxonomy expression for the profile to create a query that will return the CCE objects that match the expression. Syntax for the searches:

- **Or expression**
  ({id1} or {id2} or {id3})
- **And expression**
  ({id1} or {id2} or {id3}) and ({id4} or {id5}) and ({id6} or {id7} or {id8})
- **And expression with not**
  ({id1} or {id2} or {id3}) and not ({id4} or {id5}) and ({id6} or {id7} or {id8})
  Example:

```
<pbs:!cce module=4 class=41 searchtaxonomy="(11 or 12) and not (21
or 22 or 23)">
```

Example of search expressions:

```
19
27 or 34
(11 or 12) and (21 or 22)
(15 or 12) and not (25 or 22)
not (17 or 12) and (27 or 22)
(11 or 12 or 13 or 14) and (21 or 22 or 23) and (31 or 32 or 33 or
34)
(11 or 12 or 13 or 14) and (21 or 22 or 23) and not (31 or 32 or 33
or 34)
```

Single

Determines that a single date is sent to the system. This option operates in conjunction with the search form. Please see CCE search form. If this option is not set, you must have from and to date fields. Enter

```
Single=1
```

to supply a single date, and the system will show the objects for that date. Example:

```
<pbs:cce module=4 class=41 single=1>
```

SortOrder

Sets the sort order of the data. You can specify all available fields. For example:

```
<pbs:cce module=4 class=41 fields="Temperature" Sortorder="Temperature">
```

SortOrder will sort your data ascending by default. To reverse the order, use

```
Descending=1
```

(see above), or enter

```
(DESC)
```

after the field by which you want to sort. You can mix ascending and descending sort orders:

```
<pbs:cce ... Groups="FieldName1(DESC)" SortOrder=
"FieldName2(ASC),FieldName3(DESC),FieldName4">
```

Start

Sets the start record. This option is typically not used manually. It is used automatically by the system when you have multiple pages. Example:

```
<pbs:cce module=4 class=41 start=16 count=15>
```

TaxonomyWords	Searches for CCE objects that have the taxonomy words given in the list. It should be a list of IDs for taxonomy words: `TaxonomyWords="23,45,56"`
TaxonomyWordsAnd	This option is set to 1 if the objects should have all the taxonomy words given in the list: `TaxonomyWordsAnd=1`
ToDate	Sets the end date for the list. The date is specified in ISO format: `ToDate=20030715`
Type	Specifies the CCE type (category). This option is only used when listing one object, which belongs to a single type.Example:* * `<pbs:cce module=4 class=41 Type="CAT2901" >`
Types	Works like **Type**, but you can specify multiple types (categories). The types must be specified in a comma separated list.Example: `<pbs:cce module=4 class=41 Types="CAT2901,CAT2950">`

Reset drop-down field values

You can use a JavaScript to reset the drop-down fields to the same values that they had before the form was submitted. The following JavaScripts set the value for the fields ToDate, FromDate and Type:

> ⚠ JavaScript is case sensitive. You must write the field names exactly as they are in the form. This script must be placed at the bottom of the gen_cce_ModuleShortName.pbs template.

```
<script language="javascript">
var a = document.classfsearch.ToDate.options;
for (x=0;x<a.length;x++) {
if (("<pbs:!#ToDate#>" == "") && (a[x].value == "-1")) {a[x].selected = true;}
else
if (a[x].value == "<pbs:!#ToDate#>") {a[x].selected = true;}
}
a = document.classfsearch.FromDate.options;
for (x=0;x<a.length;x++) {
if (("<pbs:!#FromDate#>" == "") && (a[x].value == "-1")) {a[x].selected = true;}
else
if (a[x].value == "<pbs:!#FromDate#>") {a[x].selected = true;}
}
a = document.classfsearch.Type.options;
for (x=0;x<a.length;x++) {
if (a[x].value == "<pbs:!#Type#>") {a[x].selected = true;}
}
</script>
```

Objects with modules of module type Event

From version 6.2.0

There are parameters available that can filter on the special properties for Event objects:

EventFromDate	At least one event occurs on or after the given date (yyyymmdd format) (default none)
EventToDate	At least one event occurs on or before the given date (yyyymmdd format) (default none)

EventDateSort	If set to 1 the search result will be sorted by event date (default 0). It cannot be combined with other sorting parameters.
ShowEventOccurrences	If set to 0 objects appears once, on the first following event occurrence. If set to 1, the object appears once for every event occurrence. (default 0)

- CCE_ModuleShortName.pbo
- CCE_ModuleShortNameGroupHeader.pbo
- CCE_ModuleShortNameItem.pbo
- cce_ModuleShortNameNavigator.pbo
- cce_ModuleShortNameNavigatorItem.pbo
- CCE_ModuleShortNamePageLink
- CCE_ModuleShortNamePageLinkItem
- CCE_MPLACE.pbo
- CCE_MPLACEitem.pbo
- CCE-search_ovr.pbs

CCE_ModuleShortName.pbo

Contains the framework that is output once. This object file is required.

Example file name

cce_weather.pbo

Example object file

```
<html>
<body>
<table><%items%></table>
<a href="<%SortByTemperature%>">Sort By Temperature</a>
</body>
</html>
```

<%FirstHit%>	The number for the first record being viewed.
<%Items%>	Outputs the CCE items.
<%LastHit%>	The number for the last record being viewed.

<%NextLink%>	Contains a link to the next page, when not on the last page. It must be posted to the server with JavaScript like this:
	`<a href="javascript:dosubmit('<%NextLink%>');"> \\`
	`Next »</a>`

<%PrevLink%>	Contains a link to the previous page, when not on the first page. It must be posted to the server with JavaScript like this:
	`<a href="javascript:dosubmit('<%PrevLink%>');">`
	`« Previous</a>`

<%SortByDate%>	Contains a link that will sort all the objects by date. It must be posted to the server with JavaScript like this: `<a href="javascript:dosubmit('<%SortByDate%>');">Published</a>` The system will remember the sort order so you can reverse the list. To keep to the sort order when you browse using a search form, you must add a field to the form that remembers the value.
<%SortBy....%>	You can also use other fields for the sorting in addition to date. Replace "..." with the name of the field. Example: `<%SortByTemperature%>`

<%TotalHits%>	Shows the total number of records for the category being viewed.
<%search_navigator%>	The navigators build when doing search.
<%PageNo%>	Contains which pagenumber in the search result you are on (added from 6.3.0sr12)
<%PageCount%>	Contains the nunber of pages the search result consists of (added from 6.3.0sr12)
<%PageSize%>	Contains the size of a page (same as count on pbs:cce-tag) (added from 6.3.0sr12)
<%PageLinks%>	Outputs links to all pages for the curren search result. See templates cce_ModuleShortnamePageLink and cce_ModuleShortnamePageLinkItem.

CCE_ModuleShortNameGroupHeader.pbo

Contains a single field with the same name as the field specified by the **Groups** option. If you use the option

```
Groups="County"
```

you will have the field

```
<%county%>
```

in the object file. If you have specified a date field in the **Groups** option, the field will be formatted as a date if it is also listed in the **Datefields** option.

This object file is required.

Example file name

cce_weatherGroupHeader.pbo

Example object file

```
<tr>
<td>
<b>=========== <%Temperature%> ===============</b>
</td>
</tr>
```

CCE_ModuleShortNameItem.pbo

This file is output for each CCE object. This object file is required.

Example file name

CCE_WeatherItem.pbo

Example object file

```
<tr>
  <td>
    Town: <%Town%> - <%\Date%><br>
    Condition: <%Condition%><br>
    Temperature: <%Temperature%>
  </td>
</tr>
```

|| **<%Category%>** | You can use this field instead of **<%Type%>** ||

<%Categoryname%>	You can use this field instead of **<%Typename%>**
<%Class%>	Outputs the **Class** parameter from the tag.
<%Date%>	The date for the CCE object.

<%Dlink%>	A link to the detail page for the ad.
<%Expires%>	The date the ad expires.
<%FieldName%>	Outputs the number value of the extra custom field with **FieldName**.
<%ImageLink%>	A link to the image, if the ad is a display ad. If multiple images are attached to the CCE object, add the 'element' query parameter to indicate which image to display (&element=[1, 2,... n]). It must be used like this: `<img src="<%ImageLink%>&MaxW=320" border=0>`
<%LogoLink%>	A link to the logo, if the ad is a display ad. It must be used like this: `<img src="<%LogoLink%>&MaxW=320" border=0>`
<%PrintLink%>	A link to the printer friendly page. It must be used like this: `<a href="<%PrintLink%>">Print</a>`
<%OrgCCEID%> From version 6.2.0	Contains the original ID of the object. Required for allowing the user to update the object. It only has a value if the user is the creator of the object and therefore can edit it.
<%Relatedinfo%>	Outputs the information from the related field when you use the **Relatedto** function.
<%Text%>	Outputs the defined field **Text**, which is defined in the ini file.
<%TipLink%>	A link to the tip a friend page. It must be used like this: `<a href="<%TipLink%>">Tip a friend</a>`
<%Title%>	The title field for the ad.
<%Type%>	The first type for the object.
<%Typename%>	The first type name for the object.
<%UserSubmitDeleteLink%> From version 6.2.0	If the user is the creator of the item, this variable is available. It contains a link that if visited by the creator will delete the displayed object.
<%AvgRanking%> From version 6.4.0	Contains the average ranking of a CCE object.
<%RankingLink%> From version 6.4.0	The RankingLink does not contain the actual ranking value this must be added to the link by using client side scripting. The Following must be added to the link before submitting the link to the DLL '<%RankingLink%>' + '&rating=' + y Where y is the ranking given by a user.
<%TotalNoofVotes%> From version 6.4.0	Contains the total number of votes on the CCE object.

Displaying Event objects

The variables in this section are only available if the object belongs to an Event module and the <pbs:cce> parameters **EventFromDate** and **EventToDate** are used.

<%EventDate%> From version 6.2.0	Contains the first coming occurrence of the object.
<%EventDateList%> From version 6.2.0	Contains a list of the first many object occurrence dates.
<%ShowEventOccurrences%> From version 6.2.0	Contains the value of the **<pbs:cce>** tag parameter of **ShowEventOccurrences**

The following variables describe the recurrence pattern attached to the object.

<%EventFromDate%> From version 6.2.0	Contains the date from which the system is to calculate occurrences (yyyymmdd format)

<%EventToDate%> From version 6.2.0	Contains the date to which the system is to calculate occurrences (yyyymmdd format) - it is used when **StopMethod** is 0.
<%NumberOfOcc%> From version 6.2.0	Contains how many occurrences to calculate from the **EventFromDate** - it is used when **StopMethod** is 1.
<%StopMethod%> From version 6.2.0	Contains how the system is to stop calculating occurrences. 0 is on a set date, 1 is after a certain number of occurrences.
<%EventFromTime%> From version 6.2.0	Contains from what time of day the object occurs.
<%EventToTime%> From version 6.2.0	Contains to what time of day the object occurs.
<%Pattern%> From version 6.2.0	Contains the overall method used to calculate occurrences.
<%Subpattern%> From version 6.2.0	Contains the specific method used to calculate occurrences.
<%Jump1%> From version 6.2.0	Contains the first specific value in the pattern.
<%Jump2%> From version 6.2.0	Contains the second specific value in the pattern.
<%ActiveDays%> From version 6.2.0	Contains what days in the week are selected in the pattern.
<%Day1%> From version 6.2.0	If Sunday is in the **ActiveDays**, then it has value 1, otherwise it has no value.
<%Day2%> From version 6.2.0	If Monday is in the **ActiveDays**, then it has value 1, otherwise it has no value.
<%Day3%> From version 6.2.0	If Tuesday is in the **ActiveDays**, then it has value 1, otherwise it has no value.
<%Day4%> From version 6.2.0	If Wednesday is in the **ActiveDays**, then it has value 1, otherwise it has no value.
<%Day5%> From version 6.2.0	If Thurday is in the **ActiveDays**, then it has value 1, otherwise it has no value.
<%Day6%> From version 6.2.0	If Friday is in the **ActiveDays**, then it has value 1, otherwise it has no value.
<%Day7%> From version 6.2.0	If Saturday is in the **ActiveDays**, then it has value 1, otherwise it has no value.

The way a pattern is used to calculate object occurrences is described in UserSubmitFormCCERecurrencePattern.pbo.

cce_ModuleShortNameNavigator.pbo

This is the main file for outputting cce navigators.

Example file name

cce_SMSNameNavigator.pbo

Example object file

```
[%{count>1}
<td>
<form name="ccefsearch" method="post" action="/apps/pbcs.dll/cce?module=2">
<%NavigatorName%><br>
<select type="text" name="search" style="width:240px;margin:5px 0 5px 0;">
<%items%>
</select>
<input type=submit name=Go VALUE="Submit">
</form>
</td>
%]
```

Variables

<%NavigatorName%>	The name of the facet being build.
<%Items%>	Outputs the Facet elements.
<%Count%>	The number of facet elements in items.

cce_ModuleShortNameNavigatorItem.pbo

This is used for outputting each element in the navigator list.

Example file name

cce_SMSNameNavigatorItem.pbo

Example object file

```
<option value="<%ModifierValue%>" ><%Counter%> - <%ModifierName%>  (<%ModifierHits%>)
</option>
```

Variables

<%ModifierHits%>	The number of hits on this element in the navigator list.
<%ModifierName%>	The name of the element in the navigator list.
<%ModifierValue%>	The encrypted search string. Should be assigned to a post or get variable named search.
<%Count%>	The number of the current item.

CCE_ModuleShortNamePageLink

Example file name

cce_GOLFPageLink.pbo

Example object file

```
Here we have <%count%> page links: <%items%>
```

Variables

<%Items%>	Outputs the pagelink-items elements.
<%Count%>	Outputs the number of pagelinks

Note that this functionality is first available from 6.3.0sr12.

CCE_ModuleShortNamePageLinkItem

This is the main file for outputting a pagelinkitem.

Example file name

cce_GolfPageLinkItem.pbo

Example object file

```
<a href="<%PageLink%>">Page <%LinkPageNo%> of <%PageCount%>, which is items <%FirstHit%> through
<%LastHit%>. Current page has no <%PageNo%></a><br />
```

Variables

<%PageNo%>	The page number currently being shown.
<%PageCount%>	The number of pages in total
<%PageLink%>	The link to another result page
<%LinkPageNo%>	The page number for the above link

<%Counter%>	Same as <%LinkPageNo%>
<%FirstHit%>	Object number of the first item in result list for the link
<%LastHit%>	Object number of the last item in result list foir the link

Note that this functionality is first available from 6.3.0sr12.

CCE_MPLACE.pbo

CCE_MPLACEitem.pbo

CCE-search_ovr.pbs

pbs ccedetails

<pbs:ccedetails>

The CCE detail page is used to show all details about a CCE object. This page is typically linked to from the listing of CCE objects.

The tag is used like this:

```
<pbs:!ccedetails module=11 datefields="Date" dateformat="mmm dd, yyyy">
```

Or like this:

```
<pbs:!ccedetails module=4 class=41 date="20051101" type="CAT1001" id="23533221" >
```

Template

The template file for CCE object details is called **gen_cceModuleShortNamed.pbs**.

Example: **cce_weatherd.pbs**. Please notice the "d" at the end of the file name.

The template for the printer-friendly version is called **gen_cce_ModuleShortNamedp.pbs**. The printer friendly version is made for users that would like to print a CCE object. The template should contain very few graphic elements and design.

Options

Class	Determines which class of CCE objects the tags will show.
Date	The CCE object's publication date.
Datefields	Determines which of the fields in the database contain the date information.
Dateformat	Sets the date format, which is to be used when the system displays the objects.
Fields	A list of extra custom fields that should be shown.
ID	The CCE object's ID.
Module	Sets the module used by the tag. The component's short name is used to look up the templates. This option is required.Example: There is a *Weather Data* module with ID 4 and the short name *weather:* `<pbs:ccedetails module=4 class=41 date="20051101" type="CAT1001" id="23533221">`

Related	Pulls data from the related table in the database. This is typically customer information. If the related field is called CUSTOMER, you can use this option: `Related=CUSTOMER` If you use this option, the object file **CCE_ShortModuleNameRelated.pbo** is used to show the related information.
Type	The CCE object's type.

CCE_ModuleShortNameDetails.pbo

All fields in CCE_ModuleShortNameItem.pbo are also available.

<%Category%>	This field can be used instead of **<%Type%>**
<%Category01...09%>	The **<%Category01...09%>** fields can be used instead of **<%Type01...09%>**
<%Categoryname%>	This field can be used instead of **<%Typename%>**.
<%Categoryname01...09%>	Can be used instead of **<%Typename%>** fields.
<%DetailType%>	The first detail type for the object.
<%DetailType01...09%>	The additional detail types for the object.
<%DetailTypename%>	The first detail type name for the object.
<%DetailTypename01...09%>	The additional detail type names.
<%NumImages%>	Outputs the number of images for the object.
<%RelatedInfo%>	Pulls in the related information object file.
<%Type%>	Contains the first type for the object.
<%Type01...09%>	The additional types for the object.You can use `<%Type01%>, <%Type02%>` etc. for additional categories.
<%Typename%>	The first type name for the object.
<%Typename01...09%>	The additional type names.
<%SubType%>	The first subtype for the object.
<%SubType01...09%>	The additional subtypes for the object.
<%SubTypename%>	The first subtype name for the object.
<%SubTypename01...09%>	The additional subtype names.

CCE_ModuleShortNamePrintDetail.pbo

Used for the printer-friendly version of the CCE object.

The same fields as in CCE_ModuleShortNameDetails.pbo are available.

CCE_ModuleShortNameRelated.pbo

If you use the **Relatedto** option, you can use this field to show the related information. The fields that are available match the fields you have in the related table.

pbs ccefields

<pbs:ccefields>

Used to extract information from the CCE data and metadata. The tag can be used to populate the search form with relevant data.

This tag must use one or more options and can be used with the exclamation mark option.

Object files will only be used when the parameter

```
Optionlist=0
```

is used. FieldName is replaced with the value from the field option.

Options

All	Will return all types, not just ones with data attached. By default, all is not an available option. Enter `All=1` to get all types, not just the ones with data attached.
Any	Turns on *Any* as a choice in a drop-down menu. This function can only be used when you output an **optionlist**. By default, *Any* is not available as an option. Enter `Any=1` to add *Any* as a choice in the drop-down menu.
Class	Sets the class. This field must be set, if it is used. Enter `Class="x"` where x is the class.
Columns	Determines how many columns the option list will have. This option can only be used when `Optionlist=0`. The option is set like this: `Columns=3`
CustomFields	Returns all custom fields that are defined. `All=1` denoting all the fields. Example: `<pbs:!ccefields module=4 class=41 field="CustomFields" optionlist=0 any=0 all=1>`
Date	Makes the system display the dates for the days in the last week. It will also display options for the last 7, 14 and 30 days.
Dateformat	Sets the date format for date fields. Dateformat is set by using the codes found in the section Formatting dates. For example: `Dateformat="mm/dd/yyyy"`

Detailtype	Lists the detail types defined for the class. Example: `<pbs:!ccefields module=4 class=41 field="DetailType" optionlist=1 any=1>`
Excludevalues	Excludes values from the option list that is generated. By default, this option is not used. Enter a comma separated list of values that should be excluded. Example: `<pbs:!ccefields module=4 class=41 field="JOBOPTION"  excludevalues="1,2">` This tag builds a option list of the values of the extra field JOBOPTION, but will not include the values 1 and 2.
Field	Determines which field will be shown.
FieldName	Lists the values of a given extra custom field. Example: `<pbs:!ccefields module=4 field="Temperature" optionlist=1 any=1>`
Module	Determines the component from which data will be shown. This option is required.
Objectclass	Sets the objectclass. It is only valid when `Optionlist=0`

Optionlist	Turns off the output of an optionlist, and uses object files instead. The default is
	`Optionlist=1.`
	Enter
	`Optionlist=0`
	to use object files instead of an objectlist.

Related	Specifies the information that will be shown from the related information table. You can then use the related fields with the field option. Example: `<pbs:!ccefields module=4 class=41 Related="Town" field="Hotel">`
Subtype (Subcategory)	Lists the subtypes defined for the class. Example: `<pbs:!ccefields module=4 class=41 field="SubType" optionlist=1 any=1>`
Taxonomy	Lists the IDs of taxonomy words. Example: `<pbs:!ccefields module=4 field="Taxonomy" optionlist=1 any=1>`

Type (category)	Lists the categories defined for the class. Example: `<pbs:!ccefields module=4 class=41 field="Type" optionlist=1 any=1>`
Vertical	Adding the option vertical to the tag makes the tag list the options vertically instead of horizontally. The option can only be used with the **Columns** option. The option is used like this: `Vertical=1`
Week	Displays the dates for the days in the last week.

CCE_ModuleShortNameFieldsFdNmRw

CCE_ModuleShortNameFieldsFieldNameRow.pbo

When you use the **Columns** option, you must use the cce_ModuleShortNameFieldNameRow.pbo file.

Example file name: cce_weatherFieldsTypeRow.pbo

<%Col1%>	Outputs the first row.
<%Col2....9%>	You will have a col field for each row.

CCE_ModuleShortNameFieldsFieldName.pbo

Example file name: CCE_WeatherFieldsType.pbo

<%Items%>	Outputs the items.Example: `<table border=1 cellpadding=1 cellspacing=1><%Items%></table>`

CCE_ModuleShortNameFieldsFieldNameItem.pbo

This file is output per item.
Example file name: cce_weatherFieldsTypeItem.pbo

<%Id%>	The ID for the item.

<%Name%>	The name for the item. The fields can be used together like this:
	`<option value="<%Id%>"><%Name%></option>`
	Or like this:
	`<tr><td><%Id%></td><td><%Name%></td></tr>`

<%Value%>	Returns the data type defined for a custom field (see above).

pbs usersubmitform

<pbs:usersubmitform>

From version 6.2.0

The tag is used to submit data to the system through the front-end. This section will focus on submitting CCE objects.

Every CCE object specific form tag has to specify a specific Module, Class, and Type that the CCE object will belong to.

ObjectClass	Specifies what object file design class to use.
CCESection	Specifies Module, Class, and Type that the CCE object will belong to on the form "<ModuleID><*ClassID*><TypeID>". Matching settings in the publicus.ini file are required for the user to be able to do a succesful submit (see https://docs.saxotech.com/display/Onl/UserSubmitCCE+Section). The <pbs:UserSubmitForm> tag will appear in the source of the page unless the correct section exists in the publicus.ini file.
SubmitType	Specifies what type of data is to be submitted. For CCE objects this will be 'cce'. Default is 'ar' for article.

Example:

```
<pbs:usersubmitform ObjectClass="1" CCESection="2_12_Cars" SubmitType="CCE">
```

SubmitCCEObject action

This method of submitting CCE objects through the front-end is considered depricated from version 6.2.0. Instead, use the <pbs:UserSubmitForm> tag.

This method enables submitting CCE objects using the /SubmitCCEObject action. All fields, including custom fields, must be specified as content form fields when using SubmitCCEObject. The following fields must be included: Date, Module, Class, and Type.

*Examples of **Submitting a CCE-object using /SumitCCEObject action:*

```
<form name="cce" method="post" enctype="multipart/form-data" action="submitcceobject">
<input type="hidden" name="date" value="20060117">
<input type="hidden" name="module" value="4">
<input type="hidden" name="class" value="41">
<input type="hidden" name="Type" value="WTYP001">
```

The example above will only work if the form is being sent from a typical SAXOTECH Online URL. If you are sending the form from a URLRewrite page, the action must read:

```
action="/apps/pbcs.dll/submitcceobject"
```

Custom fields must be submitted in the following way:

```
<input type="FieldType" name="Field_CustomFieldName">
```

An example of submitting a custom field from the component and class listed above with the field name CONDITION could look like this:

```
<input type="text" name="Field_CONDITION" style="width : 95px" size="8">
```

Finally, all fields are submitted:

```
<input type=submit name=Go value="Submit">
```

Submitting a CCE-object with images:

```
<form name="cce" method="post" enctype="multipart/form-data" action="submitcceobject">
<input type="hidden" name="date" value="20060117">
<input type="hidden" name="module" value="4">
<input type="hidden" name="class" value="41">
<input type="hidden" name="Type" value="WTYP001">
```

The logo image should be uploaded as *Logo* input file. The first image should be uploaded as *Image01* input file:

```
<input type="file" name="Logo">
<input type="file" name="Image01">
```

Finally, all fields are submitted:

```
<input type=submit name=Go value="Submit">
```

If Date, Module, Class, or Type fields are not submitted, or are submitted with a value that does not exist, an error is returned in **<pbs:#Error#>**. The **Expires** field is optional, but the date supplied in this field will result in an error message if the format is specified incorrectly.

Options

Category	Uses a given category as a prefix instead of the standard general template.

Template

A template can be specified with a query using the template option. Otherwise a default template **gen_cce_ModuleShortNameSubmit.pbs** will be used.

The following example will use the template **gen_cce_submitphoneobject.pbs** specified in the query:

```
<form name="cce" method="post" action="submitcceobject?template=submitphoneobject">
```

The following example will use the default template **gen_cce_weathersubmit.pbs** because the ModuleShortName for component 4 is *weather:*

```
<form name="cce" method="post" action="submitcceobject?">
<input type="hidden" name="module" value="4">
```

Chapter 14 Banner Ads

The ad tag is used to pull ads from the SAXOTECH Online Ad module. This system is also used to manage ad codes from DoubleClick and other ad systems.

The ad tag requires interaction with the people selling and managing ads. You must agree on the standard sizes you are going to use for the different ad spots. If you sell ads that are larger than the space left in the templates, the ads will destroy your layout. If you use different sized ads for the same spot, ads will be randomly scaled to the size of one of the ads. This will negatively affect ad appearance.

You can have multiple ad tags on one template and multiple ads for each tag. If you have an ad tag in the header or footer, which is present on all pages, the ad system allows you to display different ads for different categories. Please see the _SAXOTECH Online User Manual _to familiarize yourself with how the ad system works.

You enter ads into the system using the SAXOTECH Online interface. For more information, see the *SAXOTECH Online User Manual.*

Flash banners
SAXOTECH Online can display Flash banner ads and create rotations of both Flash and animated GIF ads. SAXOTECH Online adheres to industry standards for rich media to ensure properly functioning and supported rich media ads. SAXOTECH Online uses the same solution for proper ad tracking as DoubleClick so third-party ads do not need to be redesigned to be tracked with SAXOTECH Online.

If you wish to serve rich media ads, you must ensure that the ads are built to the requirements specified by Adobe for allowing third-party ad serving solutions to track ads.

In order to track statistics for a Flash ad spot, Flash ads must include a special action script. The clickTAG is the tracking code assigned to an individual ad that allows SAXOTECH Online to register where the ad was displayed when it was clicked. The getURL action in this action script will navigate the browser to clickTAG by appending the variable data passed in via the OBJECT EMBED tag. Example:

```
on (release) {
```

```
if (clickTAG.substr(0,5) == "http:") {
```

```
getURL(clickTAG);
```

{code}}

```
{code}}
```

Please note that if you use swf files, they must be made with ActionScript 1.0 for Flash Player 6. The clickTAG will not work with newer versions of this scripting language.

- pbs ad
  - AdSpot.pbo
  - FlashBanner.pbo
  - LayerAdSpot.pbo
- SAXOTECH Online ads on other sites

pbs ad

<pbs:ad>

Options

Adlabel	This option only applies for sites using OAMS and is optional. It determines the Ad label to which the ad is assigned. Please see the *SAXOTECH Online User Manual* for more information.By default, the option is not used. The option is used like this: `<pbs:ad spot="TOP1" adlabel="ANCHOR">`
Adsublabel	This option applies only to sites using OAMS and is optional. It determines the Ad sub label to which the ad is assigned. Please see the *SAXOTECH Online User Manual* for more information.By default, the option is not used. The option is used like this: `<pbs:ad spot="TOP1" adlabel="ANCHOR" adsublabel="GATOR">`
Spot	Determines the ad spot for which you want to show ads. This option is mandatory and has no default value.If you put an ad in the system on TOP1, use the parameter `Spot=TOP1` to retrieve the ad. This code must be matched in the ad's management user interface.

The system does not require the use of object files. Unless you are trying to do something special, it should not be necessary to use the object files. If you need to use your own object file, it should be saved in the Objects folder root.

FlashBanner.pbo does not accept objectclass.

AdSpot.pbo

<%adimagelink%>	The image URL.
<%adlink%>	The link for the banner.
<%adlinktarget%>	The target for the link.
<%height%>	The height of the banner.
<%ilayerid%>	The ID for the ILAYER.
<%source%>	The banner's URL.

<%width%>	The width of the banner.

FlashBanner.pbo

<%clickTag%>	The link for the banner.
<%Filename%>	The banner's file name.
<%Height%>	The height of the banner.
<%Width%>	The width of the banner.

Example code

```
<OBJECT classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/
swflash.cab#version=5,0,0,0" WIDTH=<%Width%> HEIGHT=<%Height%>>
<PARAM NAME=movie VALUE="<%Filename%>[%?clickTAG=<%clickTAG$j%>%]">
<PARAM NAME=quality VALUE=high>
<PARAM NAME=bgcolor VALUE=#FFFFFF>
<EMBED src="<%Filename%>[%?clickTAG=<%clickTAG$j%>%]" quality=high bgcolor=#FFFFFF
WIDTH=<%width%> HEIGHT=<%Height%> TYPE="application/x-shockwave-flash"
PLUGINSPAGE="http://www.macromedia.com/shockwave/download/index.cgi?
P1_Prod_Version=ShockwaveFlash">
</EMBED>
</OBJECT>
```

LayerAdSpot.pbo

This file contains the same fields as **Adspot.pbo**

SAXOTECH Online ads on other sites

The following code is placed at the position where the ad will be shown:

```
<nolayer>
-snip-
</ILAYER>
```

The following code must be placed at the bottom of the page right before the

```
</body>
```

tag:

```
<LAYER>
-snip-
</LAYER>
```

Adspot	The example code displays the ad SKYSCRAPER02. To display the ads for another ad spot you must replace SKYSCRAPER02 with the correct Ad spot code.
Height and width	You must replace all height and width options with their correct values.
Spotname	The code adspot_1 identifies the spot. This code must be unique for each ad on the page. If you put another ad on the page, replace adspot_1 with adspot_2.
URL	You must change the URL to reflect the URL for your site.

Example code

At the ad position

```
<nolayer>
<iframe src="http://sh.us.publicus.com/apps/pbcsad.dll/adsource?
Spot=SKYSCRAPER02,FRONTPAGE&TARGET=_blank&H=240&W=120" name="Foo" id="Foo" width="120" height="
240" marginwidth="1" marginheight="1" scrolling="no" frameborder="0">
<a href="http://sh.us.publicus.com/apps/pbcsad.dll/adlink?Spot=SKYSCRAPER02,FRONTPAGE&ID="
target="_blank"><img src=
"http://sh.us.publicus.com/apps/pbcsad.dll/adimage?Spot=SKYSCRAPER02,FRONTPAGE" border="0"
height="240" width="120" alt=""></a>
</iframe>
</nolayer>
<ILAYER id="adspot_1" name="adspot_1" visibility="hide" width="120" height="240">
</ILAYER>
```

At the bottom of the page

```
<LAYER name="adspot_1X" src="http:
//sh.us.publicus.com/apps/pbcsad.dll/adsource?Spot=SKYSCRAPER02,FRONTPAGE&TARGET=_blank&H=240&W=12
visibility="hide" onLoad=
"moveToAbsolute(adspot_1.pageX,adspot_1.pageY);clip.height=240;clip.width=120;
visibility='show';">
</LAYER>
```

Chapter 15 Links

The Links functions allow you to build lists of links. You can use these functions to manage all types of links lists on your site.

- Links.pbo
- pbs links
- pbs linkscategorylist
    - LinksItem.pbo
    - LinksSubcat.pbo
    - Links vote form

Links.pbo

<%LinkItems%>	Outputs the items.

pbs links

<pbs:links>

You can also list links from a single category. This function is typically used to include links on a sectionfront or front page. This tag can be used on all templates.

Options

This tag has the same options as Links_Pbslinkscategorylist with the addition of two others in the table below.

Category	Sets the category ID you want to list. This option performs the same function as **maincategory** for `<pbs:linkscategorylist>`

Random	Determines whether the links should be listed in random order. By default,
	`random=0`
	Enter
	`Random=1`
	to list links in random order.

The links tag uses two object files: Links.pbo and LinksItem.pbo.

pbs linkscategorylist

<pbs:linkscategorylist>

This function lists all links in the database. Category headers are included in the list. This tag can be used on all templates.

Options

Count	Controls how many links should be retrieved. The option is set like this: `Count=30` ⚠ When using **Count,** the **Top** and **Last** arguments will be suppressed.
Last	Outputs only the newest link based on your other selections. The option is set like this: `Last=1`
Maincategory	Determines the category from which the links should be retrieved. This number is shown by the category name in the editing interface.
Objectclass	Determines which object class is used.
Region	Determines the region from which the links should be retrieved. This number is shown by the region name in the editing interface.
Sort	By entering `Sort=1` the entries will be sorted by the Sortcode. If the **Last** or **Top** parameters have been specified, Sortcode will be the secondary sort field instead of the primary one.
Subcategory	Determines the subcategory from which the links should be retrieved. This number is shown by the subcategory name in the editing interface.
Subregion	Determines the subregion from which the links should be retrieved. This number is shown by the subregion name in the editing interface.

Top	Shows the links with the highest ratings. Entering `Top=10` will result in the 10 highest rated links being shown.
Usekeywords	Used to show links with the same keywords as the story being viewed. This option can only be used on the article template. By default, this option is not used. Enter `Usekeywords=1` to use keywords for link selection.
Useobjects	Determines whether object files are used. By default, object files are not used. To enable the use of object files, add `Useobjects=1` to the tag.
Usetheme	Used to show links with the same theme as the story being viewed. This option can only be used on the article template. By default, this option is not used. Enter `Usetheme=1` to use the theme for link selection.

LinksItem.pbo

This file is output for each link.

<%Description%>	The description of the entry.
<%Gif_rating%>	A link to the rating image. This rating is given by the staff in the editing interface. The field is used like this: `<img src="<%Gif_rating%>&char=Ç&size=12&Color=0B4770" border="0" alt="" align="middle">` {{&char=«}}determines which character is used for the graphics. This function uses the Wingdings font. `&size=12` sets the size of the image. {{&Color=0B4770}}sets the color of the image.
<%Gif_userrating%>	Provides the URL to the averaged total of user-submitted votes.
<%imagelink%>	The link to the image. You can add `&Maxw=` to the field to set the image width.
<%id%>	Provides the ID of the link entry.

<%logolink%>	The link to the logo. This item must be placed inside an image source tag. You can add `&Maxw=` to the field to set the logo width.
<%\Title%>	The title.
<%URL%>	The URL for entry.

LinksSubcat.pbo

This file is output for each category and subcategory.

<%MainCategory%>	Outputs the name of the main category when it changes.
<%SubCategory%>	Outputs the name of the subcategory when it changes.

Links vote form

To allow web guests to vote on the links, you must set up a form around the list like this:

```
<form action="" method="post" name="Vform" id="Vform">
<input type="hidden" name="vote" value="0">
<input type="hidden" name="redirecturl" value="<pbs:prog>/section?Category=LINKS">
<table width="100%" border="0" cellspacing="0" cellpadding="2">
<pbs:linkscategorylist objectclass=1>
</table>
</form>
```

The form code has the following:

Form setup	`<form action="" method="post" name="Vform" id="Vform">` This line sets up the form. This form has no action, because it is handled by the JavaScript.

Hidden fields	`<input type="hidden" name="vote" value="0">`
	`<input type="hidden" name="redirecturl" value=`
	`"<pbs:prog>/section?Category=LINKS">`
	These fields are necessary to enable the vote form. The redirect URL should match the page used to view the links.

The following JavaScript must be placed in your template to enable voting:

```
<script language="JavaScript">
function DoVote(p,v) {
       var f = document.Vform;
       f.action = p;
       f.vote.value = v;
       f.submit();
}
</script>
```

In the object file **Linksitem.pbo**, you must enter the following code. This code provides the voting links:

```
<a href="javascript:DoVote('<%Votelink%>',1);">1</a>
<a href="javascript:DoVote('<%Votelink%>',2);">2</a>
<a href="javascript:DoVote('<%Votelink%>',3);">3</a>
<a href="javascript:DoVote('<%Votelink%>',4);">4</a>
<a href="javascript:DoVote('<%Votelink%>',5);">5</a>
```

Chapter 16 White Pages

The White Pages function allows you to build a White Pages register for individuals. The tags are used to pull information from the e-mail catalog.

- pbs ptemailsearch

pbs ptemailsearch

<pbs:ptemailsearch>

The <pbs:ptemailsearch> is used to display white pages. It must be placed inside a table like this:

```
<table>
<pbs:!ptemailsearch blanksearch=0>
</table>
```

Templates

The White Pages tags can be used on all templates.

Options

Blanksearch	Controls how blank searches are handled. By default, if this option is not set, or set to 1, the system will return all entries if a search without any URL parameters is performed. Enter `Blanksearch=0` to make the system return zero hits if a blank search is performed.
URL parameters	With the following parameter added to the URL, the system will search for entries in which the last name begins with H: `&Params=ENavn=H` This function is usually used for building a link block at the top of the page.

Form

The following form code sets up the search form:

```
<form name="frm" action="<pbs:prog>/section?Category=EMAIL&Form=1" METHOD="POST">
```

The category you specify sets the template that is used to display the result.

The search form can be placed on all pages. The result page is set in the form. The template that displays the result must have this tag:

```
<pbs:!ptemailsearch blanksearch=0>
```

The form uses the following fields:

Firstname	Creates the field in which the first name can be entered: `<input type="text" name="Firstname" size="15" maxlength="15">`
Lastname	Creates the field in which the last name can be entered: `<input type="text" name="Lastname" size="20" maxlength="20">`

Chapter 17 Yellow Pages

The Yellow Pages function allows you to build a Yellow Pages register for local businesses.

- pbs ypcategorylist
  - DirectoryCategoryList.pbo
  - DirectoryCategoryListItem.pbo
- pbs ypdetails
  - DirectoryDetails.pbo
  - DirectoryDetailsItem.pbo
- pbs ypsubcategorylist
  - DirectorySubcategoryList.pbo
  - DirectorySubcategoryListItem.pbo

pbs ypcategorylist

<pbs:ypcategorylist>

The tag for listing categories can be placed on all templates. Templates for listing subcategories, entries and details are specified in the tag that links to the page.

The example code used in this tag is built for use with subcategories. If your entries are placed directly on categories, you must change the link in DirectoryCategoryListItem.pbo to go to YP3 instead of YP2. This enables you to go directly to the entries when you do not have subcategories.

Options

Objectclass	Determines which object class will be used.
Useobjects	Determines whether the tag should use object files.

DirectoryCategoryList.pbo

<%items%>	Outputs the categories and must be placed inside a table:
	`<table>`
	`<%items%>`
	`</table>`

DirectoryCategoryListItem.pbo

<%Categorydetails%>	Contains the names of the subcategories.
<%CompanyCategory%>	Contains the name of the category.
<%params%>	Contains the parameters needed to link to the page. You must also specify a category to determine which template is used: `<a href="<%pbcs.dll%>/section?Category=YP2&params=<%params%>">Link</a>`

pbs ypdetails

<pbs:ypdetails>

The `<pbs:ypdetails>` tag is used to display the entries in a subcategory.

It can also be used to display all details about an entry if you do not want all information in the list. To use this function, you must link to the page from the entries list using the
`<%params%>` function. If it is used this way, you must use another object class to display the details.

The tag should be used as follows:

```
<pbs:!ypdetails useobjects=1>
```

Options

Objectclass	Determines which object class will be used.
Useobjects	Determines whether the tag should use object files.

DirectoryDetails.pbo

<%CompanyCategory%>	Contains the name of the main category in which the entries belong.
<%CompanySubCategory%>	Contains the name of the subcategory in which the entries belong.
<%items%>	Outputs the entries in the subcategory.
<%nolisting%>	Contains a value if there are no entries in the subcategory.

DirectoryDetailsItem.pbo

<%Address%>	The address.
<%Contact%>	The contact information.
<%Description%>	The description of the company.
<%\email%>	The e-mail address.
<%Link%>	The link to the homepage of the company.
<%Location%>	The city name.
<%maplink%>	The link to the MapQuest component.
<%Name%>	The name of the entry.

<%params%>	Contains the information necessary to link to a page for a single item. It must be used like this: `<a href="<%pbcs.dll%>/section?Category=YP4&Params=<%params%>">Link</a>`
<%Phone%>	The phone number.
<%State%>	The state.
<%Zip%>	The zipcode.

pbs ypsubcategorylist

<pbs:ypsubcategorylist>

The <pbs:ypsubcategorylist> tag should be used as follows:

```
<pbs:!ypsubcategorylist useobjects=1>
```

Options

Objectclass	Determines which object class will be used.
Useobjects	Determines whether the tag should use object files.

DirectorySubcategoryList.pbo

<%Categoryname%>	Contains the name of the category for which you are listing subcategories.
<%items%>	Outputs the subcategories.

DirectorySubcategoryListItem.pbo

<%CompanySubCategory%>	Contains the name of the subcategory.
<%params%>	Contains the parameters needed to display the detail category. This field is used in the link like this: `<a href="<%pbcs.dll%>/section?Category=YP3&params=<%params%>">Link</a>`

Chapter 18 Guides

Guides are functions that let you build a cross between the yellow pages and the links database. It is also possible to add additional fields. The tag for listing the guide categories can be placed on all pages. The template used for listing the entries for the categories is called **gen_guide.pbs**.

- Guides free text search
  - SearchResultGuide.pbo
  - SearchResultGuideItem.pbo
- pbs ptguides
  - GuideSubcatItem.pbo
- pbs ptguidesearchitem
  - GuideSearchItem.pbo
  - GuideSearchItemExtraItem.pbo
- pbs ptguidesearchlist
  - GuideBusinessHours.pbo
  - GuideSearchList.pbo
  - GuideSearchListHeader.pbo

    - GuideSearchListItem.pbo
- pbs ptguidesearchoptions
    - GuideSearchOptionsItem.pbo
    - GuideSearchOptionsRow.pbo
- Guide search form

Guides free text search

This section covers free text search for the guides.

Template

This search form can be placed on all pages.

Form code

This line sets up the search form:

```
<form action="search?
*Type=guide*
&category=guide" method=post>
```

You must put the option

```
Type=guide
```

in the tag. The category option is optional.

Fields

Crit	Used to search all the other fields.
Guide_address	Used to search the address field.
Guide_city	Used to search the city field.
Guide_description	Used to search the description field.
Guide_email	Used to search the e-mail field.
Guide_name	Used to search the name field.
Guide_phone	Used to search the phone field.
NoBlankCheck	Used to determine that searches can be performed with blank search criteria.

Search result files

The search result is displayed on the template **gen_guidesearch.pbs**. This template has the same features as the regular search template. For information about the tags, please see Result page.

SearchResultGuide.pbo

SearchResultGuide.pbo

This object file has the same fields as SeachResult.pbo. Please see SearchResult.pbo.

SearchResultGuideItem.pbo

<%Address%>	The item's address.
<%City%>	The item's city name.
<%Contact%>	The item's contact information.
<%Counter%>	Countains the counter value.
<%E-mail%>	This tag is written with a hyphen and contains the email address of the item.

<%Fax%>	The item's fax number.
<%ItemLink%>	The link to the guide item.
<%Name%>	The item's name.
<%Phone%>	The item's phone number
<%URL%>	The link to the item's homepage.

pbs ptguides

<pbs:ptguides>

This function lists all guide categories. You can filter the list of categories using the listed options.

Template

The tag for displaying the main list can be placed on all pages.

Options

These options are not mandatory.

GetActiveAds	Enabling this option ensures that only categories with active ads are listed. For example: `<pbs:ptguides maincatfilter=A subcatfilter=F getactiveads=1>`

Maincat	Enables you to select the top level to which you would like to limit the listing. The easiest way to find this number is to view the category link in the full listing. By default, all categories are listed. Enter
	`Maincat=XX`
	to list only the category with number
	`XX`

MainCatFilter	Allows you to filter the list of categories based on the first character in the category. This parameter can also be placed in the URL. This option is not used by default. Enter `MainCatFilter=A` to list only the categories that start with "A."
Noall	When present, Noall prohibits the output of a link to the top level of a category.
SubCatFilter	Allows you to filter the list of categories based on the first character in the subcategory. This parameter can also be placed in the URL. This option is not used by default. Enter `SubCatFilter=A` to list only the subcategories that start with "A."

GuideSubcatItem.pbo

<%GuideSubCat%>	The name of the subcategory.
<%GuideSubCatLink%>	The link to the subcategory.
<%GuideDetCat%>	The name of the detail category.
<%GuideDetCatLink%>	The link to the detail category.
<%GuideDetCatid%>	The detail category ID.
<%GuideSubCatid%>	The category ID for the subcategory.

pbs ptguidesearchitem

<pbs:ptguidesearchitem>

You can also use this tag to display the information for a single item. The details page for one entry automatically uses the same template and tag as the main list of entries. The only difference is that it uses another set of object files. When a web guest links to the details page, the system will use this URL:

```
/apps/pbcs.dll/ptshowguideitem
```

Template

The information is usually displayed on the **gen_guide.pbs** template. You can also use this tag on other templates.

Options

ItemID	Used to specify the ID of the item being shown. It can also be specified in the URL. This option should only be used when you are hard coding the tag to show a single entry.

GuideSearchItem.pbo

<%Address%>	The entry's address.
<%AltPhone%>	The alternative phone number for the entry.
<%Contact%>	The contact person.
<%Counter%>	Holds the counter information for the current record.
<%Description%>	The description of the entry. **Showref** must be >0 for this field to be available.
<%E-Mail%>	This tag is written with a hyphen and contains the email address.
<%ExtraNoTranslate_X%>	Retrieves the extra field with ID X (untranslated) from the database.
<%Extra_X%>	Retrieves extra field X (translated) from the database.
<%Extras%>	Outputs the extra fields for the entry. This field uses GuideSearchItemExtraItem.pbo.
<%Fax%>	The fax number.
<%GuideDetailCat%>	The name of the detail category to which the entry belongs.
<%GuideDetailCatId%>	The detail category ID.
<%GuideId%>	The ID for the record.
<%GuideMainCat%>	The name of the main category to which the entry belongs.
<%GuideMainCatId%>	The main category ID.
<%GuideName%>	The name of the entry as a link.
<%GuideSubCat%>	The name of the subcategory to which the entry belongs.
<%GuideSubCatId%>	The subcategory ID.

<%Imagelink%>	The URL for the attached image. To resize this image, you must use the link like this: `<img src="<%Imagelink%>&Maxw=200" border=0>` In this example, the image will be displayed with a maximum width of 200 pixels.
<%Location%>	The location for the entry.
<%Logolink%>	The URL for the logo. To resize the logo, you must use the link like this: `<img src="<%Logolink%>&Maxw=200" border=0>` In this example, the logo will be displayed with a maximum width of 200 pixels.
<%Name%>	The name of the entry as a link if the link field has content.
<%Phone%>	The phone number for the entry.
<%Sortcode%>	Has the value from the sortcode field on the current record.
<%TipLink%>	The link to the tipform. It uses the same template as the article tip function. The e-mail sent to the user uses GuideTipsText.pbo.
<%Url%>	The entry's homepage.

GuideSearchItemExtraItem.pbo

Output for all the extra fields attached to an entry.

<%Name%>	The name of the extra field.
<%Value%>	The content of the extra field. This field is formatted by the system.
<%ValueNoTranslate%>	Retrieves the information (untranslated) from the database. If you have HTML code in the extra field, this code will be output exactly as entered.

pbs ptguidesearchlist

<pbs:ptguidesearchlist>

Template

The template **gen_guide.pbs** is used to display entries when you link from the list of categories. If you specify the main category and subcategory, the tag can be used on all pages.

URL

The URL for using the gen_guide template is **/apps/pbcs.dll/ptshowguide**

Options

Count	Determines how many entries should be displayed per page.

Maincat	Specifies the main category. It should not be used on gen_guide.pbs. By default, all categories are listed. Enter
	`Maincat=X`
	to show only the category with ID
	`X`

Namesort	Allows you to sort a main category, ignoring the subcategories and detail categories. This option is not used by default.The function is enabled by entering `Namesort=1` in the tag.
Objectclass	Sets the object class for the list.
OnlyShowGuidesWithActiveAd	Allows you to select only guides that have an active ad (the ad has a start date before today and end date after today).
Show ads	Shows ads associated with the guide. Uses GuideAdItem.pbo. Enter `ShowAds`
ShowBusinessHours	Makes the **GuideOpeningHours** variable available on the GuideSearchListItem.pbo.

Subcat	Specifies the subcategory. It must be used with the **Maincat** option. This option should not be used on the gen_guide.pbs template. By default, all categories are listed. Enter
	`Subcat=X`
	to show only the subcategory with ID
	`X`

GuideBusinessHours.pbo

Used with the GuideSearchListItem.pbo.

CloseTime	The time the business closes.
DayName	The name of the week day.
OpenTime	The time the business opens.

GuideSearchList.pbo

<%Category%>	The category of the guide.
<%DetCat%>	The detail category of the guide.
<%FirstHit%>	The number for the first hit being viewed.
<%GuideDetailCat%>	The name of the detail category.

<%GuideMainCat%>	The name of the main category.
<%GuideSearchDetails%>	Lists all the items from the GuideSearchListItem.pbo
<%GuideSubCat%>	The name of the subcategory.
<%LastHit%>	The number for the last hit being viewed.
<%Location%>	The county/kommune for the guide.
<%MainCat%>	The ID of the main category, as taken from the tag.
<%MainCat2%>	Used to get ID of other categories of guides from the tag.
<%MainCat3%>	Used to get ID of other categories of guides from the tag.
<%NextLink%>	A link to the next page if you are viewing a multipage listing and you are not on the last page.
<%PreviousLink%>	A link to the previous page if you are viewing a page that is not the first page in a multipage listing.
<%SubCat%>	The ID of the subcategory, as taken from the tag.
<%SubCat2%>	Used to get ID of other subcategories of guides from the tag.
<%SubCat3%>	Used to get ID of other subcategories of guides from the tag.
<%TotalHits%>	The number of total hits for the list being viewed.

Filtering

This block of code gives you a drop-down menu that will filter the entries in the guide by the start letter of each of the entries' names. This code should not be changed.

```
<form action="<%PostLink%>" method="GET">
<input type="Hidden" name="Category" value="GUIDE">
<input type="Hidden" name="MainCat" value="<%MainCat%>">
<input type="Hidden" name="SubCat" value="<%SubCat%>">
<input type="Hidden" name="First" value="1">
<input type="Hidden" name="Number" value="<%Number%>">
<select name="Filter"><option value="">None</option><%filteroptions%></select>
<input type="submit" value="Filter">
```

Drop-downs

If you want the values of the drop-down to be remembered when you submit a search, you must add the following block to the bottom of GuideSearchList.pbo:

```
<SCRIPT>
var a = document.guidesearch.SubCat.options;
for (x=0;x<a.length;x++) {
 if (a[x].value == "<%SubCat%>") {a[x].selected = true;}
}
var a = document.guidesearch.Location.options;
for (x=0;x<a.length;x++) {
 if (a[x].value == "<%Location%>") {a[x].selected = true;}
}
var a = document.guidesearch.extra_508.options;
for (x=0;x<a.length;x++) {
 if (a[x].value == "<%Extra_508%>") {a[x].selected = true;}
}
document.guidesearch.filter.value = unescape("<%Filter$j%>");
</SCRIPT>
```

With extra fields

To keep the information about the extra fields, the next and previous links must look like this:

```
<a href=
"<%PreviousLink%>[%&extra_508=<%extra_508%>&SearchExtras=508,515%][%&musthaveextras=<%musthaveextr
>Previous</a>
<a href=
"<%NextLink%>[%&extra_508=<%extra_508%>&SearchExtras=508,515%][%&musthaveextras=<%musthaveextras%>
>Next</a>
```

GuideSearchListHeader.pbo

Output before the list of items.

<%GuideItemHeader%>	Outputs the name of the category.

GuideSearchListItem.pbo

This list can display the same fields as the detail page, except the extra fields. The detail page is described later in this chapter.

<%Address%>	The address of the customer in the guide item.
<%AltPhone%>	An alternate telephone number for the customer in the guide item.
<%Category%>	The name of the current category. This does not contain the category of the guide item being output; it contains the value of the page's category query parameter.
<%CountyID%>	The ID of the county/kommune for the customer.
<%Counter%>	Holds the counter item.
<%Description%>	A description of the customer from the guide item.
<%DetCat%>	The name of the detail category.
<%E-Mail%>	The customer's email address from the guide item.
<%Fax%>	The customer's fax number from the guide item.
<%GuideAdItems%>	The result of the GuideAdItem.pbo.
<%GuideOpeningHours%>	The business opening hours, as defined in GuideBusinessHours.pbo
<%IsLastItem%>	This field has the value 1 when the record is the last record in the list.
<%ItemID%>	Shows the database ID of the guide item.
<%ItemLink%>	Used to link to the detailed information.
<%Location%>	The customer's city, state, and zip code or post code from the guide item.
<%MainCat%>	The main category of the guide item.
<%Name%>	The name of the guide item.
<%NewDetailCat%>	This field has the value 1 if the detail category has changed. This allows you to use the following code to output a header between the different items: `[%{NewDetailCat=1<b><%DetailCat%></b> %]`
<%Phone%>	The customer's phone number from the guide item.
<%SortCode%>	An integer that is calculated when the guide is created to determine how the guide item will be sorted.
<%SubCat%>	The subcategory of the guide item.This only has a value of the subcat option is set in the pbs:ptguidesearchlist tag.

pbs ptguidesearchoptions

<pbs:ptguidesearchoptions>

This tag is used to list a single subcategory or detail category. The tag is used like this:

```
<pbs:ptguidesearchoptions type=subcategories maincategory=111 noall>
```

Template

This tag can be used on all templates

Options

Columns	Enables the listing of entries in columns. By default, columns are not used. Enter
	`Columns=X`
	to have
	`X`
	number of columns.

Empty	Enter `Empty=0` to make the tag list just those categories with content.
Maincategory	Determines which category will be listed. This option has no default value.To select a category, use `Maincategory=X` where X is the category number. This category number is shown in the editing interface.
Noall	When present, Noall prohibits the output of a link to the top level of a category.
Notempty	Makes the tag list only those categories with content.
Objectclass	Sets the object class for the object files.
Sortcode	To list the guides sorted in ascending order by the SortCode field enter: `Sortcode=1` Please note that the list will be sorted by guide name after sort code.

Type	Determines whether to list subcategories or detail categories. This option has no default value.To list subcategories, use
	`Type=subcategories`
	To list detail categories, use
	`Type=detailcategories`
	To list locations, use
	`Type=locations`
	To list guide names, use
	`Type=guidenames`
	To list counties, use
	`Type=counties`
	To list "extra," use
	`Type=extra`

Vertical	Outputs vertical columns instead of horizontal columns.

GuideSearchOptionsItem.pbo

<%Link%>	Contains the link to the listing of entries.
<%Name%>	Contains the name of the subcategory or detail category.

Example

```
<a href="<%link%>"><%name%></a><br>
```

GuideSearchOptionsRow.pbo

This file is used when you use the **Columns** option in the tag.

<%col1%>	Outputs the first column when the **Columns** option is used in the tag.
<%col2...9%>	You must have the same number of column fields as you have set in the option.

Guide search form

It is possible to build an interactive search page. This section also covers how you can search by extra fields.

Template

Guide_guide.pbs

This example uses the template guide_guide.pbs. If you are going to use another template name, you must change the action for the form as well.

This line sets up the form. The name of the form must be "guidesearch":

```
<form name="guidesearch" action="ptshowgiude?Category=GUIDE" method=post>
```

These hidden fields set the main category to search:

```
<input type="Hidden" name="
*MainCat*
" value="2">
<input type="Hidden" name="
*SearchExtras*
" value="508,515">
<input type="Hidden" name="
*Extra_515*
" value="<pbs:#extra_515#>">
```

Two extra fields are also defined:

- **Extra_508** is used for the rating and is remembered by the drop-down.
- **Extra_515** is used for featured restaurants. It is used to remember the value.

City	Used to select the region to search. `<select name="Location" size="1">` `<pbs:ptguidesearchoptions maincategory=2 type=locations>` `</select> `

Close form	This block closes the form and shows the matching entries:
	`</form>`
	`<pbs:ptguidesearchlist count=15>`

Cuisine	Used to build the category drop-down. It lists all the subcategories for the dining guide. `<select name="SubCat" size="1">` `*<pbs:ptguidesearchoptions maincategory=2 type=subcategories>*` `</select> `

Filtering	This block is used to switch the filtering on the featured restaurants. It uses JavaScript to output different text depending on whther you are showing all restaurants or just the featured ones:
	
	`<a href="javascript:Extras();">View`
	`<script>`
	`Function Extras()`
	`{`
	`if (document.guidesearch.Extra_515.value=="1")`
	`{`
	`document.guidesearch.Extra_515.value = "";`
	`}`
	`else {`
	`document.guidesearch.Extra_515.value = "1";`
	`}`
	`document.guidesearch.submit();`
	`}`
	`if ("<pbs:#Extra_515#>"=="1") {`
	`document.write("all");`
	`}`
	`else {`
	`document.write("featured");`
	`}`
	`</script>`

Name	This field line is used for free text search by the restaurant's name. `<input type="text" name="filter" size=20> `

Review	Used to search the extra field for the review. The option values must be coded to match the value entered on the items.
	
	`<select name="extra_508" size="1">`
	`<option value="">All</option>`
	`<option value="0">1 star</option>`
	`<option value="1">2 stars</option>`
	`<option value="2">3 stars</option>`
	`<option value="3">4 stars</option>`
	`<option value="4">5 stars`

Submit	This line submits the search form. You can use a regular submit button instead. `<a href="Javascript:document.guidesearch.submit();">Search</a>`

Chapter 19 Q&A

The functions described in this chapter can be used for both polls and contests. They are set up through the SAXOTECH Online interface. For more information about polls and contests, please see the SAXOTECH Online User Manual.

Please note that you cannot change the number of answers after a poll has accumulated data. Doing this breaks the poll.

- pbs pollarchive
  - PollArchiveItem.pbo
- pbs polldata
  - PollDataResults.pbo
  - PollDataResultsItem.pbo
  - PollDataResultsItemEntry.pbo
  - PollForm.pbo
  - PollFormAlternatives.pbo
  - PollFormAlternativesItem.pbo
  - PollFormItem.pbo
  - Polls_Error.pbo with pbs polldata

        - Polls_Saved.pbo with pbs polldata
        - Poll Statistics
    - pbs pollmini
        - Pollmini.pbo
        - PollMiniChart.pbo
        - PollMiniNoPoll.pbo
    - pbs quest_multipage
        - MultiPageQuestIndex.pbo
        - MultiPageQuestPage.pbo
        - MultiPageQuestPageX.pbo
        - MultiPageQuestSubmit.pbo
        - MultiPageQuestWrongAnswer.pbo
        - Multipage self test
        - MultiQuestAltItem.pbo
        - Poll_Error.pbo with pbs quest_multipage
        - Poll_Saved.pbo with pbs quest_multipage
    - pbs questform
        - Polls_Error.pbo with pbs questform
        - Polls_Saved.pbo with pbs questform
        - QuestForm.pbo
        - QuestFormAlternativesItem.pbo
        - QuestFormItem.pbo

pbs pollarchive

<pbs:pollarchive>

The poll archive is viewed by accessing a URL like this:

```
/apps/pbcs.dll/section?Category=POLLARCHIVE*&ShowCategory=OPINION*
```

In this example, you need to make a template called pollarchive_ovr.pbs. ShowCategory in the URL determines the category from which polls are shown.

Options

Chatheight	Determines how high the graph will be in pixels.
Chartwidth	Determines how wide the graph will be in pixels.
Count	Controls how many polls should be shown.
Dateformat	This is how the dates will be presented. This tag uses the regular date format options.

PollArchiveItem.pbo

<%ChartLink%>	The link to the chart.
<%EndDate%>	The end date for the poll.
<%Ledetekst%>	The description of the poll.
<%PollID%>	The ID code for the poll. This is used in a hidden field.
<%Question%>	The question that was asked.
<%StartDate%>	The start date for the poll.
<%Votes%>	The total number of votes.

pbs polldata

<pbs:polldata>

This function is used for a full page poll. To access this poll, you must link to **`/apps/pbcs.dll/poll?Category=Pollcategory`** where Pollcategory is the category in which the poll is placed. It is used on the template **gen_poll.pbs**. After the poll has been posted, the template gen_poll.pbs will be used to show either the thank you message or an error message.

Options

Chartlegend	Determines whether and where a chart legend should be output. The default position is to the right of the graph. The possible options are: Top, Bottom, Left, Right and Off. For example, enter `chartlegend=Bottom` in the tag.
Charttype	Specifies the chart type that will be used. By default, the chart type is a bar chart. The following chart types are available: Pie, Statsbar, Planner, Barstack, Line, Chart, Bar, Areastack, Mline, Tinystat and Tinystatsbar.
Chartvalues	Determines whether the system should write the percent and number of votes directly into the graphic. By default, the number of votes is printed next to each bar. The following options are available: off, number, percent and both.
Imgwidth	Specifies the width of the image showing the graph. By default, the system size is used. Enter `imgwidth=X` to make the graph X pixels wide.
Imgheight	Specifies the height of the image showing the graph. By default, the system size is used. Enter `imgheight=X` to make the graph X pixels high.
Labeltrim	Used to shorten labels, so you can better control the layout. By default, labels are shown as they are entered. Enter `labeltrim=20` to cut off the labels after 20 characters, for example.
Objectclass	Sets the object class. By default, the object class is not set. Enter `objectclass=x` in the tag to set the object class. Replace x with any number between 1 and 999.
Rgbgraphcolors	Defines the colors used for the graph. If you specify three and have five alternatives, the colors will loop, so option four will have the same color as option 1. The system uses the default colors. Enter `rgbgraphcolors=EE3300,33EE00,00EE33` to specify three graph colors.

Showresults	Determines whether web guests will be taken directly to a results page or to a thank you page. The default is to go to the thank you page. Entering
	`showresults=1`
	takes guests to the results page. Entering
	`showresults=0`
	takes guests to the thank you page.

Useobjects	Determines whether object files are used. By default, object files are not used. Enter `useobjects=1` in the tag to use object files.

PollDataResults.pbo

<%Items%>	This field lists all the items.
<%\Title%>	The poll title.

PollDataResultsItem.pbo

<%chart%>	The URL for the chart. If you want to disable the chart's 3D appearance, you must use a regular expression like this: `<%chart$r("(&3d\x3d1)=&3d\x3d0")%>`
<%entries%>	Outputs the entries.
<%question%>	Contains the question.
<%totalvotes%>	Shows the total number of votes.

PollDataResultsItemEntry.pbo

<%entryname%>	Shows the name name of the entry.
<%numberofvotes%>	Contains the number of votes.

<%votepercent%>	Contains the percent of votes for the entry - without "%".Here is an example of an PollDataResultItemEntry.pbo file:
	`<tr>`
	`<td width="120">`
	`<font face="Verdana, Arial, Helvetica, sans-serif" size="2">`
	`<%entryname%></font></td>`
	`<td width="170">`
	`<img alt="" src="/img/reddot.gif" width="<%votepercent%>%" height="10"></td>`
	`<td align=right><font face="Verdana, Arial, Helvetica, sans-serif" size="2">`
	`<%votepercent$f("0")%>%</font></td>`
	`</tr>`

PollForm.pbo

<%Action%>	The URL to which the form must be posted.
<%ID%>	The ID code for the poll. This is used in a hidden field.
<%Info%>	The information field from the poll
<%Items%>	Contains all the poll items.

<%MaxAnswer%>	The maximum answer code for the poll. This is used in a hidden field. The Name in this field must be
	`"max":`
	`<input type="hidden" name="max" value="<%MaxAnswer%>">`
	Please note that the above line is required in the pbo-file.

<%Redirect%>	The redirect URL for the poll. This is used in a hidden field.
<%\Title%>	The title of the poll

PollFormAlternatives.pbo

<%Items%>	Lists all the alternatives.
<%Name%>	Contains the name.
<%type%>	Contains the poll type.

PollFormAlternativesItem.pbo

<%Alternative%>	Contains the alternative text.
<%Name%>	Contains the question ID.
<%Value%>	Contains the alternative ID.

PollFormItem.pbo

This file is output for every question in the poll.

<%Item%>	Used with the **Required** field to keep track of the question.
<%Question%>	Contains the question.
<%QuestionField%>	Lists all the alternatives.
<%Required%>	Used with the **Item** field to keep track of the question

Polls_Error.pbo with pbs polldata

This page is used if the user does not fill in the name/address form correctly, or does not fill in a required field.

<%backlink%>	Contains the link back to the poll page.

Polls_Saved.pbo with pbs polldata

This page is shown after the poll has been saved.

<%chartlink%>	Contains the link to the results page.

Poll Statistics

The full page poll confirmation page automatically links to this page. It uses the same tags and templates as the full page poll. If you want to link to the page manually, you must use this URL:

`/apps/pbcs.dll/pollstats?Category=POLL&`**`ID=12`**

You must specify the ID for the poll you want to show.

pbs pollmini

<pbs:pollmini>

The Mini poll is the most common version of the poll tag. It can be used on all pages. In modern browsers, this poll function allows the user to vote without refreshing the entire page.

The tag can look something like this:

```
<pbs:pollmini category=FRONTPAGE labels="trailing" width="121" height="135"
imgwidth="115" imgheight="60" bgcolor=e9eef9>
```

This tag uses an ilayer for Netscape 4.X and an iframe for Internet Explorer and other browsers that support iframe.

To get this function to work in old browsers, you must place the tag `<pbs:#PollLoad#>` at the bottom of the page right before the `</body>` tag. You can put this tag in your standard footer so it is always included.

Options

Bgcolor	Defines the poll's background color. An iframe is a separate HTML document with its own **<body>** tag. You must write the parameter like this: `bgcolor=#e9eef9`
Category	The category in which the poll is placed. If this option is not specified, the current category is used.
Current	Makes the tag display the poll for the current date, and not the storyÕs date. This option should be present in the tag. Otherwise, web guests will see an old poll when they read an old story.
Height	The height of the Iframe or Ilayer. It should be in the range 100 to 400 pixels, depending on your design.
Imgheight	The height of the result graph shown to people that have voted. It is also shown after the poll is closed.
Imgwidth	The width of the result graph shown to people that have voted. It is also shown after the poll is closed.
Objectclass	Sets object class. By default, this option is not set. Enter `objectclass=x` to set the class where x is any number between 1 and 999.

Profile	The profile attached to the poll. If this option is not specified, the current profile is used. For example will this tag filter on current category and the profile of 1002:
	`<pbs:pollmini current profile="1002">`
	The following tag on a section front, **/section?Category=Sports** where the Sports category has an attached profile, 1004:
	`<pbs:pollmini current>`
	will filter on Sports category and on profile 1004.

RGBgraphcolors	Determines which colors the graph bars should have. This option uses the RGB color model. The option is set like this: `rgbgraphcolors="9999cc,cc9999,99cc99"`
Width	The width of the Iframe or Ilayer. It should be in the range 100 to 400 pixels, depending on your design.

This tag uses two object files: One for showing the poll and one for showing the result.

Pollmini.pbo

<%entry1%> to <%entryXX%>	These are the possible entries. By putting [and] at the beginning and end of a line, you indicate that the line should only be output if it contains data.
<%NoPoll%>	Has a value if there is no poll available. It can be used to show a message informing the web guest that no poll is available.
<%posturl%>	The URL to the program that processes the submission.
<%question%>	Outputs the question.
<%systemfields%>	Outputs the hidden system fields that the poll needs to function.
<%\Title%>	Outputs the poll title.

PollMiniChart.pbo

<%chart%>	The link to the chart.
<%htmlchart%>	The chart in HTML table format. This option uses the following pbo files to render results: PollDataResults.pbo, PollDataResultsItem.pbo, and PollDataResultsItemEntry.pbo.
<%question%>	Outputs the question.
<%votes%>	The number of votes cast.

PollMiniNoPoll.pbo

Used to output a message if there is no poll available. It is only used when the flag for always using iframe is set in Publicus.ini. The use of this file is automatic. If you want to use the

```
<%NoPoll%>
```

option in Pollmini.pbo, this file cannot be present for the object class.

pbs quest_multipage

<pbs:quest_multipage>

This function uses one page per question instead of having all questions on a single page. To access this contest, you must link to **/apps/pbcs.dll/quest?Category=Questcategory** where Questcategory is the category in which the contest is placed.

This function uses the **gen_quest.pbs** template. This quiz type requires another tag to be put on the gen_quest.pbs template. You should make a template for the category because this function uses a different tag than the normal contest.

This tag gives you a lot of freedom in designing the contest, because it uses object files.

The multipage contest needs the following object files.

MultiPageQuestIndex.pbo

The start page for the contest.

<%Info%>	Outputs the information field for the contest.
<%Link%>	The link to the next page. It should be placed inside a link tag like this: `<a href="<%Link%>">Start</a>`
<%Title%>	Outputs the title field for the contest.

MultiPageQuestPage.pbo

<%Alternatives%>	This is where the alternatives are output. This tag should be placed inside the form.
<%CurrentAnswers%>	Holds the current answers. It must be present and used like this: `<input type=hidden name="CurrentAnswers" value="<%CurrentAnswers%>">`

Form	The basic form code must include at least the following information.
	`<form name="frmOppgave" action="<%Link%>" method=POST>`
	`<input type=hidden name="CurrentAnswers" value="<%CurrentAnswers%>">`
	`<input type=hidden name="oppg" value="1">`
	`<input type="submit" name="Submit" value="Submit">`
	`</form>`

Freetext answer	It is also possible to have free text answers. To enable this function, use the following code: `<textarea name="answer" value="" cols=50 rows=10></textarea>` This question type is typically used in questions of the type "In twenty words or less, tell us why you should win." You use this code instead of the **<%alternatives%>** field and typically only on one of the questions.
<%Link%>	The link to the next page. It must be placed inside the action parameter of the form.
<%Question%>	Outputs the question.
<%QuestionNumber%>	Outputs the question number.

MultiPageQuestPageX.pbo

It is possible to design the layout of each question. To do this you must add a number corresponding to the question number after the file name. If you want to make a special design for question number 3, the file must be MultiPageQuestPage3.pbo.

You use the same field names as the regular question page.

MultiPageQuestSubmit.pbo

Form	All input fields must be inside the following form:
	`<Form  action="<%Link%>" method=post>`
	`Place fields here`
	`</form>`

Hidden fields	The following hidden fields must be present in the form and not changed:
	`<input type=hidden name="max" value="<%NumberOfQuestions%>">`
	`<input type=hidden name="quest" value="1">`
	`<input type=hidden name="chart" value="0">`
	`<input type=hidden name="id" value="<%ID%>">`
	`<%Answers%>`

Required fields	Use the following hidden field to specify which fields must be completed. `<input type="hidden" name="requiredfields" value="name,address1,,city,state,zip">` You can make all fields required.

User information fields	The following fields can be used to get information about the user.
	`<input type=text name="Title">`
	`<input type=text name="Name">`
	`<input type=text name="Address1">`
	`<input type=text name="Address2">`
	`<input type=text name="City">`
	`<input type=text name="State" size="2">`
	`<input type=text name="Zip" size="5">`
	`<input type=text name="Age">`
	`<input type=text name="Birthdate">`
	`<input type=text name="Email">`
	`<input type=text name="Phone1">`
	`<input type=text name="Phone2">`
	`<input type=text name="Cellphone">`
	`<select name="Sex">`
	The last field should be a selection box.

MultiPageQuestWrongAnswer.pbo

This version of the contest uses the same object files as the pbs MultipageSelftest. In addition, it has one extra object file. It is used when the user answers a question incorrectly.

This tag uses object files. In addition to the object files used for multipage contest, the following file is used:

<%RestartLink%>	This variable can be used to link back to the beginning of the contest.
<%TryAgainLink%>	This variable can be used to link back to the question that was not answered correctly.

Multipage self test

Multipage self test is a variation of a multipage contest. Instead of the web guest providing his name and number, he will be presented with a page showing how many questions were answered correctly.
The results page uses the object file MultiPageQuestSubmit.pbo to show how many correct answers the web guest had:

MultiPageQuestSubmit.pbo

<%AllCorrect%>	This field has a value if all questions were answered correctly.
<%CorrectAnswers%>	Contains the number of correct answers.
<%NumberOfQuestions%>	Contains the number of questions.

MultiQuestAltItem.pbo

Used to output the alternatives.

<%AltAlfaNumber%>	Outputs a letter corresponding to the alternative number, where A is the first alternative and B is the second.
<%AltNumber%>	Outputs the alternative number. It is used in the input selection like this: `<input type="radio" name="answer" value="<%AltNumber%>">`
<%AltText%>	Outputs the alternative text.

Poll_Error.pbo with pbs quest_multipage

This page is used if the user does not fill in the name/address form correctly. You can place a JavaScript on the page so guests can go back and complete the form. The script could look like this:

```
<a href="javascript:history.go(-1)">Go back</a>
```

Poll_Saved.pbo with pbs quest_multipage

This page contains no variables. It is used to thank the web guest for taking part in the contest.

pbs questform

<pbs:questform>

It is possible to use the Polling component for contests. Depending on the type of contest, different templates will be used. *Single* and *grouped* refer to options in the editing interface. The tag must be placed on the `gen_quest.pbs` template or a template for a specific category. After the poll has been posted, the template gen_poll.pbs will be used to show either the thank you message or an error message.

To access the contest, you must link to `/apps/pbcs.dll/quest?Category=Questcategory` where Questcategory is the category in which the contest is placed.

Options

Category	Determines the category from which contests will be shown.

Polls_Error.pbo with pbs questform

This page is used if the user does not fill in the name/address form correctly. You can place a JavaScript on the page so guests can go back and complete the form. The script could look like this:

```
<a href="javascript:history.go(-1)">Go back</a>
```

Polls_Saved.pbo with pbs questform

This page contains no variables. It is used to thank the web guest for taking part in the contest.

QuestForm.pbo

This object file is not needed by the system. It is optional. You can use this object file to change the layout of the contest.

In addition to the fields below, you need to have a submit button and the fields for capturing name and address. These fields are described in the section MultiPageQuestSubmit.pbo.

<%Info%>	The information field.
<%questformitems%>	Displays the item file.
<%\Title%>	The contest's title.

QuestFormAlternativesItem.pbo

Defines the alternatives in the question form. It can only be used with the radio button option.

<%Alternative%>	Displays the alternative text.The fields are typically used like this: `<input name="<%Name%>" type="radio" value="<%Value%>"> <%Alternative%> `
<%Name%>	Used to remember the question number.
<%Value%>	Used to remember the alternative number.

QuestFormItem.pbo

This object file is not needed by the system. It is optional. You can use this object file to change the layout of the contest.

<%alternatives%>	Outputs the alternatives.
<%question%>	Outputs the question.
<%Name%>	The name field.
<%Address1%>	The first address field.
<%Address2%>	The second address field.
<%EMail%>	The email address field.
<%Age%>	The age field.
<%Sex%>	The gender field.
<%Phone1%>	The primary phone number field.
<%Phone2%>	The secondary phone number field.
<%Title%>	The tfitleield.
<%MiddleName%>	The middle name field.
<%Birthday%>	The birthday field.
<%CellPhone%>	The cellphone number field.
<%City%>	The city field.
<%State%>	The state field.
<%Zip%>	The zip code field.
<%US%>	The country field.
<%CustomField1%>	A custom field defined by you.
<%CustomField2%>	A custom field defined by you.
<%CustomField3%>	A custom field defined by you.

Each of the variables corresponds to a poll property field, and will be set to 1 if the field is active on the poll. Also, all of the above variables can be appended `Required` to determine that the field is required. For example,

```
CellPhoneRequired=1
```

if the CellPhone field is required.

Chapter 20 Events

The Events tags are used to build the pages that display the events calendar. There are also tags to build the pages that let web guests

input events into the system.

- Printer friendly calendar
- pbs events_previous_link
- Events setup
- Events search form
- Add events form
- pbs eventlist
    - EventDateHeader.pbo
    - EventDetails.pbo
    - EventItem.pbo
    - EventLocationHeader.pbo
    - EventSubjectHeader.pbo
- pbs events_next_link
- pbs cinemalist
    - MovieList.pbo
    - MovieListItem.pbo

Printer friendly calendar

make a printer friendly calendar, you must perform the following steps.

Create the template

Create the template **print_events.pbs**. This template should not contain any graphic elements or other elements that cannot be printed.

Example template:

```
<h1>Printed from the Times Gazette Inquirer</h1>
<pbs:eventlist local=1 showdate=0 todaydate=1 datesort=0>
```

Add JavaScript

The following JavaScript must be added to the JavaScript section on the main events template:

```
function DoPrint () {
 var f = document.searchform;
 f.action = "<pbs:prog>/events?category=print";
 f.submit();
}
```

This JavaScript requires that your search form is called `searchform`.
If your search form has another name, you must replace `searchform`
with the name of your form to enable this function.

Add link

You must also add the following link somewhere on your main events template:

```
<a href="javascript:DoPrint();">Print this calendar</a><br>
```

pbs events_previous_link

<pbs:!#events_previous_link#>

A navigation tag, this outputs the link to the previous page:

```
<pbs:!#events_previous_link#>
```

Must have same cache setting (!) as pbs:eventlist.

Events setup

The following steps must be taken before you can use the Events Module.

Publicus.ini settings

You must configure the Publicus.ini based on the codes that have been assigned to your site. Make the following settings in the GENERAL section:

Fylke

- Redfylke specifies your main region.
- EventsClass specifies additional regions.
- Sets the class for your subject codes. The class must match the class from the database.
- If you do not want to share your events with other sites, you must make the OnlyLocalCommunities setting:

```
OnlyLocalCommunities=1
```

- Specify the class for the editing interface in the AGENDACLASSSUBJECTMAPPING section.

Create regions

Regions can only be created by a database administrator. Unless you are sharing your region codes with other sites, you must use a unique code for "Fylker." Both "Fylker" and "Kommuner" must be created.

Create subjects

Subjects can only be created by a database administrator. Unless you are sharing your subjects with other sites, you must use a unique code for the events class.

Events search form

The search form is used to select different events. This form can be placed on all pages. The results will be displayed on the **gen_events.pbs** template.

The <pbs:eventsearch> tag creates the correct form action based on the category being viewed.

The <pbs:eventoptions> tag is used to extract the available options for the events drop-downs.

Options

Communities	Used when listing communities. If OnlyLocalCommunities is specified in the Publicus.ini, the set of counties (Fylke) will be limited to the set specified in the Publicus.ini. For example, Fylke=01,02...Only counties/communities with registered events on the site are displayed. `<pbs:eventoptions communities default="">` By default, this sets the default community number. This community will be selected in the drop-down when the page loads.
Filtersubjects	Used with **Subjects**, this is removes subjectcodes from the drop-down menu. The subjects must be specified in a semicolon separated list.
Intervals	Used when listing intervals. It selects which interval should be set as default. `<pbs:eventoptions intervals intextended selectdate default="7">`
Intextended	Used with **Intervals,** this enables the extended select criteria.
Selectdate	Used with **Intervals** this option must be present if you are going to use the pop-up calendar for date selection.
Sub	Used with **Subjects** this option enables the display of subcategories.

Subjects	Used when listing subjects. It selects the default subject in the drop-down. The subject ID must be specified. `<pbs:eventoptions subjects default="0" filtersubjects=5 Sub>` The option EventsClass in the Publicus.ini controls which subjects are listed.

The `<pbs:cinemaopts>` tag is used to list the cinemas.

```
<pbs:cinemaopts all type=REGIONCINEMAS>
```

Options

All	Enables the *All cinemas* option at the top of the list.
Type	Selects the data set. Contact your system administrator for the correct value.

Variable tags

These tags are used to set the correct value for input fields after the form has been submitted. The **Eventoptions** tag manages this function for drop-downs.

<pbs:#DateSelected#>	Remembers the value of the select date field.
<pbs:#FreeText#>	Remembers the value of the free text search field.

Building the form

The following fields are needed or can be used to build a search form.

The form must start with the following line:

```
<form name="searchform" action="<pbs:eventsearch>" method="post">
```

Action	The action must be `<pbs:eventsearch>`.
Category	(Hidden field) Sets the template name for the results page. For example, the value business will result in the template **business_events.ovr** being used. This field is optional. If the value is not set, the current category will be used. `<input type="hidden" name="Category" value="BUSINESS">`

Community	Enables the drop-down used to show regions with registered events.
	`<select name="`
	`*Community*`
	`">`
	`<pbs:eventoptions communities>`
	`</select>`

Datesort	(Hidden field) Enables sort by date. `<input type="hidden" name="DateSort" value="1">`

Intervals	Lets the web guest select the date or range for which she wants to list events.
	`<select name="`
	`*Interval*`
	`" OnChange="`
	`*javascript:ResetDate*`
	`();`
	`">`
	`<pbs:eventoptions intervals intextended selectdate default="7">`
	`</select>`

The JavaScript must be present if you are using the **Select date** function.You can also use the function pbs showdate to output dates in the drop-down. This is used instead of the eventoptions tag. To do this, put the following line inside the `<select></select>` tags:

```
<option value="<pbs:showdate format=yyyymmdd offset=17>">
<pbs:showdate format="ddd dd.mm" offset=17>
</option>
```

This will output an option field for the date 17 days into the future.

Name	The field name must have the value `"searchform"` for the JavaScripts used later to work.
OnlyLocalEvents	(Hidden field) Used to filter out events that are not registered on the same site as the page is viewed. `<input type="hidden" name="OnlyLocalEvents" value="1">`

Subjects	Enables the drop-down box for subjects:
	`<select name="Subject">`
	`<pbs:eventoptions subjects subcategories>`
	`</select>`

Pop-up calendar

This function allows web guests to pick a date from a pop-up calendar.

Fields	You must have the two following fields:
	`<input type="text" name="dateselected" value="<pbs:#DateSelected#>" size="8" maxlength="8"`
	`"{width: 5em}">`
	`<input name="DateChoice" type="button" value=".." onClick="javascript:choosedate();">`
	The textfield must be named `"dateselected"` and have the value `<pbs:#DateSelected#>`.
	The button used to open the calendar must be named `"DateChoice"` and have the onClick JavaScript.

JavaScript	This JavaScript is used by the pop-up calendar. The pop-up calendar itself is described in the section Pop-up calendar.
	
	`<script language="javascript">`
	`function choosedate () {`
	`window.open('/g/kalender_eng.html?searchform.dateselected','','menubar=0,titlebar=0,width=`
	`document.searchform.Interval.options[document.searchform.Interval.options.length-1].select`
	`}`
	`function ResetDate () {`
	`if (document.searchform.Interval.selectedIndex != (document.searchform.Interval.options.le`
	`{`
	`document.searchform.dateselected.value = "";`
	`}`
	`}`
	`</script>`
	
	This JavaScript should not be changed.

Interval drop-down	The interval field must have the **onChange** field and the eventoptions tag must have the **selectdate** option like this:
	
	`<select name=`
	`*"Interval"*`
	`{{ }}`
	`OnChange=`
	`*"javascript:ResetDate()*`
	`;`
	`">`
	`<pbs:eventoptions intervals intextended`
	`*selectdate*`
	` default="7">`
	`</select>`

Browse by calendar

Use the tag pbs calendarto build a calendar into the page that allows web guests to browse the events calendar by clicking on the date. In the CalendarDateItem.pbo, you must use the following link:

```
<a href="events?Category=EVENTS&Date=<%ISODate%>&CalDate=<%ISODate%>">
```

Free text search

```
<INPUT name="freetext" size="10" maxlength="50" value="<pbs:#FreeText#>" style="width: 170px">
```

This field allows web guests to perform free text searches in the events. The name of the field must be free text. If you set the value for this field to

```
<pbs:#freetext#>,
```

the search text will be remembered from search to search.

The other fields will limit your search. If you have determined that you only want to display one day and enter a search string, only that day will be searched. You should build a separate free text search form.

Example code:

```
<form name="event2" action="<pbs:eventsearch>" method="post" style="margin-left: 0;
margin-right: 0; margin-top: 0; margin-bottom: 0;">
<input type="hidden" name="OnlyLocalEvents" value="1">
<input type="hidden" name="Interval" value="120">
<INPUT name="freetext" size="10" maxlength="50"  style="width: 170px">
<input type="Submit" value="Go">
</form>
```

Cinema

This drop-down is used to list all cinemas:

```
<select name="
Cinema
" class="dropdown">
<pbs:cinemaopts all type=REGIONCINEMAS>
</select>
```

The drop-down must be named

```
"Cinema"
```

Using cookies

This option creates a cookie, which is used for storing the user's search settings between each visit to the page. The function is optional. If you choose to use the function, all of the following fields and scripts must be present.

JavaScripts	The scripts used by the function are loaded using the following line: `<script language="JavaScript" src= "/apps/cookiesearchparams.js" type= "text/javascript"> </script>`
Initializing cookie	The following lines load the stored cookie. `<script Language="Javascript"> InitCookieSearchParams(document.searchform) </script>`
Setting cookie	The form must also be modified so the cookie is set when the form is submitted: `<form action="<pbs:eventsearch>" method="post" name="searchform" *onSubmit="pbsSetCookie(this)* " style="margin-top: 0; margin-bottom: 0;">`
Cookiename	The following field sets the name of the cookie. `<input type="hidden" name="cookiename" value= "events">`

Add events form

It is possible to let web guests add events to the system. These events are not published directly, but placed in a queue so staff can review them first.

Link to the form

To link to the registration form, use the following link:

```
<a href="eventsadd?Category=XXX">Add your event</a>
```

If you have not enabled user registration for the category, the user will be presented with the registration form. If user registration is enabled, the user must first log in. Login should be required on the category, so you capture the name, email address, and phone number for the person registering the event.

After logging on, the user will be sent to the page for adding an event. Please see Chapter 25 User Registration.

This page uses the **gen_eventsadd.pbs** template.

Form

The following lines are needed to set up the form:

```
<form action="<pbs:eventsave>" method="post" enctype="multipart/form-data">
All the fields go here
<input type="submit" value="Post event">
</form>
```

System fields

Other system fields	Every input field has a matching variable, which remembers the information if the submission of the form fails. The variable is listed with the field.
Published	(Hidden field) Determines whether the agenda is published automatically. `<input type="hidden" name="Publisert" value="1">`
Redirect	(Hidden field) Defines the page to which the web guest is directed after submitting the form. `<input type="hidden" name="RedirectURL" value="/misc/thankyou.htm">`

Submitter	(Hidden field) Sets the submitterÕs name.
	`<input type="hidden" name="Submitter" value="<pbs:#events_submitter#>">`
	`<input type="hidden" name="required" value="FromDate,Fromtime">`
	`<input type="hidden" name="Class" value="0">`

System messages	This variable outputs system messages. These messages are output if the registration of a new event fails. `<pbs:#events_message#>`

User input fields

Active days	The following checkboxes are used to select active days. This is used for recurring events that do not happen every day of the week. This info must be insted if you will use ToDate.
	`<input type="Checkbox" name="aday" value="1" <pbs:#events_aday1#>> Sunday`
	`<input type="Checkbox" name="aday" value="2" <pbs:#events_aday2#>> Monday`
	`<input type="Checkbox" name="aday" value="3" <pbs:#events_aday3#>> Tuesday`
	`<input type="Checkbox" name="aday" value="4" <pbs:#events_aday4#>> Wednesday`
	`<input type="Checkbox" name="aday" value="5" <pbs:#events_aday5#>> Thursday`
	`<input type="Checkbox" name="aday" value="6" <pbs:#events_aday6#>> Friday`
	`<input type="Checkbox" name="aday" value="7" <pbs:#events_aday7#>> Saturday`

Extra info	This field is used to add extra information. This can be information that only appears in your print calendar.
	`<textarea name="ExtraInfo" cols=40 rows=5 wrap="VIRTUAL" style="width: 250px;">`
	`<pbs:#events_ekstrainfo#>`

From date	Captures the start date. `<input name="FromDate" type="text" size=8 maxlength=8 value="<pbs:#events_fradato#>" >`
From time	Captures the start time. `<input name="Fromtime" type="text" size=4 maxlength=4 value="<pbs:#events_frakl#>" >`
Link 1 text	Used to add the description for the link. `<input name="Linktext1" type="Text" size="40" maxlength="50" style="width: 250px;">`
Link 1 URL	Used to add a link to the event. `<input name="Link1" type="Text" size="40" maxlength="50" style="width: 250px;">`
Link 2 text	Used to add the description for the second link. `<input name="Linktext2" type="Text" size="40" maxlength="50" style="width: 250px;">`
Link 2 URL	Used to add another link to the event. `<input name="Link2" type="Text" size="40" maxlength="50" style="width: 250px;">`
Location	The location of the event. `<input name="Location" type="Text" size=40 maxlength=50 style="width: 250px;" value="<pbs:#events_sted#>" >`
Mailing address	The mailing address for the event. `<input name="Mailingaddress" type="Text" size=40 maxlength=50 style="width: 250px;" value="<pbs:#events_postadresse#>">`
Phone	The contact phone number. `<input name="Tlf" type="Text" size=40 maxlength=50 style="width: 250px;" value="<pbs:#events_tlf#>">`
Photo	Used to upload a photo on the event. `<input name="Picture" type="file" id="Picture" size="26" style="width: 250px;">`

Region	Selects the region.
	`<select name="Kommune" style="width: 250px;">`
	`<pbs:eventoptions regkommuner default="2001">`
	`</select>`

Street address	The address for the event. `<input name="Streetaddress" type="Text" size=40 maxlength=50 style="width: 250px;" value="<pbs:#events_gateadresse#>">`

Subject	Determines the subject. You must use the pbs tag to build the list of valid subjects:
	`<select name="Subject" style="width: 250px;">`
	`<pbs:eventoptions regemner>`
	`</select>`

Title	The title of the event. `<input name="Title" type="text" size=40 maxlength=50 style="width: 250px;" value= "<pbs:#events_tittel#>" >`
To date	Captures the end date. `<input name="ToDate" type="text" size=8 maxlength=8 value="<pbs:#events_tildato#>" >`
Totime	Captures the end time. `<input name="Totime" type="text" size=4 maxlength=4 value="<pbs:#events_tilkl#>" >`

pbs eventlist

<pbs:eventlist>

This is the main tag for building the overview of events. This tag can be used on all templates. However, if you are going to use the search form, it must be placed on the template **gen_events.pbs**.

This tag uses object files if they exist. If you have a search form for the classified ads on the same page as the listing, the options set in the form will override the options in the tag. You can also put parameters in the URL. These will override the form.

Appearance options

Datesort	Determines whether the entries are sorted by date or by region. By default, entries are sorted by region.Enter `datesort=1` to sort the entries by date.

Infowinheight	Determines the height of the info window in pixels. For example:
	`Infowinheight=400`
	If you leave this option blank
	`(infowindoheight="")`
	the JavaScript window will not be auto created.

Infowinwidth	Determines the width of the info window in pixels. For example: `Infowinwidth=700`
Objectclass	Sets the object class.

Pagelimit	Limits the number of events shown per page. This option should only be used on locked lists of events. If you show multiple days and have the **Pagelimit** option set, the navigation becomes illogical. By default, all events matching the criteria are shown. Enter
	`pagelimit=X`
	to show only
	`X`
	events per page. The link to the next and previous pages will change to say next and previous X hits.

Showdate	Enables date headers in the events list. This option can only be used if **Datesort** is enabled. This option is turned off by default. Enter `showdate=1` to show date headers.
Todaydate	Disables the output of the date for events that happen today. By default, the date header for today's events is enabled. Enter `todaydate=1` to disable the date header for today's events.
ShowMoreInfo	Retrieving a list of events would generate a javascript function that was used in the <%InfoLink%> Url. Now the <pbs:eventlist> tag has a parameter that dictates if the variable and js function are to be outputted. `ShowMoreInfo=0` default is 1

Content options

Limit the search when you load the page. If you have a search form for events on the same page, the options in the search form will override the parameters in the tag when the form is submitted. Parameters in the URL will override the search form.

Community	Controls which regions will be shown. By default, all regions will be shown. By setting this value, you can select one or multiple regions. This option is set this way: `community="4002,4003"` This will select the regions with the specified region codes. The region codes are shown by the region name in the SAXOTECH Online interface. The communities must be in quotes and separated by commas.
Current	This option makes the list ignore the date in the URL and use the current publishing day. This option should be used when you list events on story pages and section fronts with stories. Otherwise, the list of events will be for the story's publishing day.

Detail	Selects a subcategory for the specified subject. By default, all items on subcategories are shown. Enter
	`detail="10"`
	to show the subcategory with the ID
	`10`
	The **Subject** must also be specified for this to work.

Filtersubjects	Used to remove specified subjects from the list. By default this option is not used. Enter `filtersubjects="1,3"` to NOT show subjects 1 and 3. The values must be in quotes and separated by commas.

Interval	This option sets the number of days shown in the list. The default is one day. This option can be set from one to 120 with
	`interval=X`
	where
	`X`
	is the number of days.

Local	Determines whether only local events will be shown. This option can also be set in the Publicus.ini. If you use the option in the tag, you will override the ini file setting. By default the setting in the Publicus.ini is used. If this is not set, all events will be shown. Enter `local=0` to show all events. Enter local=1 to show only local events.

Published	Overrides the published option so that unpublished events are listed. Only published events are shown. Enter
	`published=0`
	to show unpublished events instead. Enter
	`published=-1`
	to show all events. The published flag is ignored.

Subject	Controls which subjects will be shown. By default, all subjects will be shown.By setting this value, you can select a single subject: `subject="12"` The subject codes are shown by the subject name in the SAXOTECH Online interface.

Other options

Cinema	Extracts events from a single cinema. By default, the event list shows events from all providers. Enter
	`cinema=XX`
	where
	`XX`
	is the cinema code. The event list will only show events for one provider. The cinema codes are most easily found by viewing the source of the user editing page. The cinema code is the combination of your site code plus the cinema ID. Example:
	`cinema=GE36`

Guideid	Used to show events for a single entry from the guide database. For this option to be used, it must be turned on in the Publicus.ini. It is not used by default. This option is typically used when you have an events tag inside GuideSearchItem.pbo. You can also use it inside a macro and read the guide ID value from the URL.You can use the following code both inside guidesearchitem.pbo, and when you have the field
	`guideid=X`
	in the URL where
	`X`
	is the guide ID you want to show.
	`<pbs:eventslist guideid="<%Guideid%>">`

Ignoremovie	Determines whether information from cinemas should be treated as movie listings or as normal events. If you are using the cinema function to let other event providers manage their own events, this option must be used. When the option **Ignoremovie** is enabled, events from cinemas are treated as regular events.This option has no parameters. If the option is not present, the cinema events are treated as movies.
People	Used to display information registered in Agenda People. It is used to list birthdays, anniversaries, and so on. You must set the option **AgendaPeopleClass** in the Publicus.ini to match the class you are using. You can get this value from the system administrator. This option is not used by default. Enter `people=1` to use information from Agenda People instead. The templates are called "AgendaPeople" instead of "Event,", so the item file is called **AgendaPeopleItem.pbo** instead of **EventItem.pbo**. The system will use the standard templates as a fallback.

URL parameters

Most of the options you can set in the tag can also be set as parameters in the URL. This is done by adding

```
&Params=
```

to the URL. Here are the most common functions:

Date	To set the start date, use the following code: `/apps/pbcs.dll/events?category=EVENTS&*Dato=20030726*`

Interval	To set the interval in the URL, use the following code: `/apps/pbcs.dll/events?category=EVENTS&Params=*Interval=30*`
Region	To set the region, use the following code: `/apps/pbcs.dll/events?category=EVENTS&Params=*Kommune=4003*`
Subject	To set the subject, use the following code: `/apps/pbcs.dll/events?category=EVENTS&Params=*Emne=1*`

EventDateHeader.pbo

This file is output when the date changes.

<%\Date%>	The date that the event happens.

EventDetails.pbo

This page must be built as a complete HTML document. It is not wrapped in the gen_events.pbs template.
This object file can use all variables from the other object files.

<%\Date%>	The date that the event happens in English.
<%Dato%>	The date that the event happens in Danish.
<%NetInfo%>	Contains the same information as the field `<%Netinfotext%>` in EventItem.pbo.

EventItem.pbo

<%bannerimagelink%>	Outputs the link to the banner image. It must be used with the banner URL.
<%bannerurl%>	The link for the banner image.
<%CommunityName%>	The name of the region in which the event is placed.
<%Date%>	The date that the event happens or the first date the event happens if it is recurring.
<%Daterange%>	The from/to date value if the event occurs on multiple days. If the event occurs on a single date, this field is blank otherwise.
<%Datetime%>	This variable is used to display both date and time.
<%Days%>	The names of the days on which the event happens, if it happens on multiple dates.
<%ExtraInfo%>	Corresponds to the **ExtraInfo** (for print) in the editing interface. This field can be used for extra information that should only appear in the print edition of the newspaper.
<%ID%>	This is the ID for the event item.

<%Imagelink%>	Contains the link to the image. It must be used like this: `<img src="<%ImageLink%>&MaxW=150">` &Maxw= sets the maximum image width.
<%Infolink%>	Contains a link to EventDetails.pbo. It can be used with the field **Netinfo** to only output the link if this field contains information. `[{netinfo=1}|]\\` `[|]<a href="<%Infolink%>">More info</A> %"linktype="raw"` `wikidestination="" originalalias="%{netinfo=1}<a href="<%Infolink%>"` `>More info</A> %" class="linkerror" >%{netinfo=1}<a href="` `<%Infolink%>">More info</A> %`
<%InfoLinkNoWin%>	Contains the link to the details for the event. A new window is not opened. If you are using <%InfoLinkNoWin%> then you should add following parameter to the url: <%InfoLinkNoWin%>&ISODate=<%IsoDate%>. If you won't the event date and not todays date.
<%IsoDate%>	The date in ISO format.
<%link1%>	Contains custom link 1.
<%link2%>	Contains custom link 2.
<%linktext1%>	Contains the link text for custom link 1.
<%linktext2%>	Contains the link text for custom link 2.
<%Location%>	This is used for the venue name.
<%Mailingaddress%>	The event's mailing address.
<%maplink%>	This field contains the map link for MapQuest. This function requires that MapQuest is installed on your web server and that you have a contract with MapQuest.
<%MapURL%>	Contains the link to the map for customers using TTmaps. Not used by MapQuest.
<%Netinfo%>	Contains the Netinfo text.
<%netinfotext%>	This field has the value 1 if the field **Netinfo** in the editing interface contains information. This field is used with the **Infolink** field to open a window containing the complete event information.
<%Phone%>	The phone number.
<%Streetaddress%>	The event's street address.
<%Text%>	This field is used for the description of the event.
<%Time%>	The from/to time for the event.
<%Title%>	This is the title for the event.

EventLocationHeader.pbo

This file is output when the location changes.

<%CommunityName%>	The name of the community.

EventSubjectHeader.pbo

This file is output when the subject changes.

<%Subject%>	The event type field.
<%subsubject%>	This option has no parameters. If the option is not present, the events are treated as movies.

pbs events_next_link

<pbs:!#events_next_link#>

A navigation tag, this outputs the link to the next page:

```
<pbs:!#events_next_link#>
```

Must have same cache setting (!) as pbs:eventlist.

pbs cinemalist

<pbs:cinemalist>

This tag is used to display movies through the Events Module. This functionality is not related to the Movies Module. The tag can be used on all templates.

MovieList.pbo

<%Cinema%>	The name of the cinema if you are listing only one cinema.
<%Items%>	Contains all the items.
<%NextLink%>	The link to the next page.
<%NoItems%>	This field has a value if there are no items to list.
<%PreviousLink%>	The link to the previous page.

MovieListItem.pbo

<%Body%>	The movie description.
<%Cinema%>	The name of the cinema.
<%ComingAttraction%>	Contains information about whether the movie is shown now or in the future. It is a Boolean variable.
<%Country%>	The production country.
<%\Date%>	The start date.
<%EndDate%>	The end date for the listing.
<%EndTime%>	The end time for the listing.
<%Genre%>	The genre for the film.
<%Homepage%>	The link to the movie's homepage.
<%ID%>	The ID for the movie.
<%Image%>	The link for the image.
<%ImgHeight%>	The image height.
<%ImgWidth%>	The image width.
<%ISODate%>	The start date in ISO format.
<%ISOEndDate%>	The end date for the listing in ISO format.
<%\Language%>	The language.
<%Link1%>	Contains a link. This could be to a cinema showing the movie, a movie review, and so on.
<%Link2%>	Contains a link. This could be to a cinema showing the movie, a movie review, and so on.
<%LinkText1%>	Contains text referring to the content of **<%Link1%>**.
<%LinkText2%>	Contains text referring to the content of **<%Link2%>**.
<%MovieReview%>	The same as <%Link1%>.

<%OnlineOrder%>	Contains information about whether tickets can be ordered online.
<%PlayTime%>	The running time.
<%Rating%>	The movie's rating.
<%Region%>	The region.
<%Shows1%>	Show times for Sunday.
<%Shows2%>	Show times for Monday.
<%Shows3%>	Show times for Tuesday.
<%Shows4%>	Show times for Wednesday.
<%Shows5%>	Show times for Thursday.
<%Shows6%>	Show times for Friday.
<%Shows7%>	Show times for Saturday.
<%ShowsInfo%>	Contains various information about the movie. For example, parental guidance regarding the movie.
<%ShowsToday%>	Show times for today.
<%StartTime%>	The start time for the listing.
<%\Title%>	The title of the movie.
<%Trailer%>	The link to the trailer.

Chapter 21 Forums

Forums give you the ability to interact with your web guests. This function allows guests to comment on stories and sections.

There are several settings in the Publicus.ini that must be added before you can use forums on your site. Please see the FORUM section .

- Message filtering
- Message form
  - ForumNewMessage.pbo
  - ForumNewMessageExtLogin.pbo
  - ForumNewMessageUnavailable.pbo
- pbs Captcha
  - Captcha.pbo
- pbs forumoptions
  - ForumPreferences.pbo
  - ForumPreferencesSaved.pbo
  - ForumPrivateEmail.pbo
  - ForumPrivateEmailMessage.pbo
  - ForumPrivateEmailSent.pbo
- pbs news_content
  - ForumMessage.pbo
  - ForumMessageAttachmentItem.pbo
  - ForumMessageThreadItem.pbo
  - ForumThemeItem.pbo
  - ForumThemes.pbo
  - ForumThemesGroupItem.pbo
- pbs news_count
- pbs news_grouplink
  - ForumGroupLink.pbo
- pbs news_groups
  - ForumGroupHeadlineItem.pbo
  - ForumGroupItem.pbo
  - ForumGroups.pbo
- pbs news_hotthreads
  - ForumHotThreadItem.pbo
  - ForumHotThreads.pbo
- pbs news_latest
  - ForumLatestMessageItem.pbo
  - ForumLatestMessages.pbo
- pbs news_messages
  - ForumMessageItem.pbo
  - ForumMessages.pbo
  - ForumMessagesMessageThreadItem.pbo
  - ForumNavBar.pbo

    - ForumNavBarItem.pbo
- pbs search
- pbs searchresult and forums
    - SearchResultForum.pbo
    - SearchResultForumItem.pbo
- pbs userverify
    - UserVerificationMail.pbo
    - UserVerify.pbo

Message filtering

The forums support message filtering. This function is used to replace profanities and other objectionable content. The system matches whole words.

The file containing the list of blocked words is called **nolist.txt** and is placed in the root folder of the website. Enter one word per line. Phrases can also be used. The file should be no longer than 50 words.

This system is not foolproof. If your users want to use these words they will find a way around it. By replacing letters with numbers and misspelling words is possible bypass the system like this: `r3allybadw0rd`.

Message form

This form is used for all message creation and editing.

ForumNewMessage.pbo

<%PostMessageLink%>	The link to the program that processes the submission.<form method="POST" action="<%PostMessageLink%>">
<%Attachments%>	Lists file attachments. Please see ForumMessageAttachmentItem.pbo.

<%AttachmentsNotImage%>	The number of non image attachments associated with this message. This is the same is
	`*<%Attachments%>*`
	minus
	`*<%Images%>*`

<%\Author%>	The name of the logged on user. This is usually the nickname, but if there is no nickname it is the user name.
<%AvatarLink%>	An image link to the logged on user's avatar.
<%Body%>	The body text of the current message.
<%CanPost%>	This field has a value of 1 if the user has a user level above 1.
<%DeleteMessageLink%>	A link to an action that will cause the top message to be deleted with all its sub messages. Deleting messages actually only marks those messages as unpublished.
<%DeleteReplyLink%>	A link to an action that will cause the current message to be deleted with all its sub messages. Deleting messages actually only marks those messages as unpublished.
<%Description%>	The group name for this message
<%Edit%>	This field has a value of 1 if editing an existing message.
<%EditMessageLink%>	The link to the page for editing the current message.
<%\Email%>	The logged on user's email address.
<%ExtEmailAdr%>	When extended login is used, this is the user's email address.
<%ExtFirstName%>	When extended login is used, this is the user's first name.
<%ExtSurName%>	When extended login is used, this is the user's surname.
<%FirstName%>	The logged on user's first name.

<%FromUrl%>	The referring URL.
<%GroupID%>	The ID of the group in which this message will be created.
<%GroupLink%>	A link to the list of messages for a link.
<%GroupOptions%>	This field contains an HTML option fragment that contains a list of group names and IDs for for the theme in which the current message exists. This is typically used to present a drop-down list of groups that users can use to navigate to a new group.
<%HasNew%>	This has a value of 1 when the current message has replies that have been added since the current user's last visit date/time.
<%HotThread%>	This field has the value 1 if the thread is hot. The **Hotthreshold** parameter on the pbs news_content tag controls the number of replies required before a message is considered hot.
<%Images%>	(Threaded Forums only) The number of image attachments associated with this message.
<%InReplyTo%>	The ID of the message to which this post responds.
<%IsGrpLocked%>	This has a value of 1 if the group in which this message belongs is locked.
<%IsModerated%>	The group in which this message exists is moderated.
<%IsModerator%>	This field has a value of 1 if the logged on user is a moderator.
<%IsMsgLocked%>	This has a value of 1 if the message is locked.
<%IsNew%>	This has a value of 1 when the current message has been created since the current user's last visit date/time.
<%ISOCreated%>	The same as <%Created%> except in ISO format.
<%ISOLastMessageDate%>	The same as <%LastMessageDate%> except in ISO format
<%ISOPublished%>	The same as <%Published%> except in ISO format.
<%IsSticky%>	This has a value of 1 if the message is a sticky message. A sticky message will appear above all normal messages.
<%LastMessageDate%>	The date/time the last message in the thread was created.
<%LastName%>	The logged on user's last name.
<%Link%>	The link to the full text message.
<%MessageID%>	A value the represents the current message as a number.
<%Name%>	The user name of the logged on user.
<%NewMessageLink%>	The link to the page for posting a new top level message.
<%NickName%>	The logged on user's nickname.
<%PostMessageLink%>	A link that will submit the message to the system.
<%PreviewBody%>	A preview of the body with all the graphics resolved.
<%PreviewMessageLink%>	A link that will preview the message.
<%Published%>	For moderated forums, this is the date/time the moderator published the message. For non-moderated forums this is the date/time the message was created.
<%QuoteBody%>	The quoted body of the original message.
<%ReplyMessageLink%>	The link to the page for posting a reply.
<%ShowAllLink%>	A link to a list of messages for this messages group that is not restricted by **MaxDays.**
<%ShowRepliesLink%>	The same as <%Link%>
<%Signature%>	The logged on user's signature.
<%Theme%>	The theme name for this message.
<%ThemeDescription%>	The theme description for this message.
<%ThemeId%>	The ID of the theme in which this message will be created.
<%ThemeLink%>	A link to the list of groups for the theme.

<%ThreadCount%>	The number of replies to this message.
<%\Title%>	If this is a reply, this field contains the title of the original posting with Re: appended to the beginning.
<%Today%>	This field has a value of 1 when there is a reply to the current message that has been created today.
<%ToggleLockLink%>	A link that will cause the messages locked status to toggle. A locked message will become unlocked and an unlocked message will become locked.
<%UserLevel%>	The user level of the logged on user.
<%ValidCaptcha%>	When a forum is protected using a captcha tag, if the user submits an incorrect solution the dll returns the preview page instead. <%ValidCaptcha%> is only available when the solution was incorrect. It has a value of 0.

Output form fields

The following form fields are sent to the application to complete the action following a form submission.

Attachments	Any HTTP file input type `<input type="file" ...>` will be interpreted for attachments.
Author	The author name that is used in non Extended Forums.
Body	Body text of the new message.
DeleteAttachments	If this item has a checked value, then any current attachments will be removed from the message.
Edit	If this value is set to 1, an existing record is to be updated.
Email	The e-mail address that is used in non Extended Forums.
EmailReplies	If this value is set to 1, the user is to be notified when someone has replied to the message.
Extra_XXX	Name and value details can be stored against a message. These details will be displayed in the editing interface when editing the message. An example:<input type="hidden" name="Extra_postcode" value="GL7 3QU">
GroupRedirect	If this value is set to 1, all the messages for the current group will be displayed after the post completes.
InReplyTo	The message ID against which a reply is set.
MessageID	The ID of the message being edited. See Message form.
ModeratorMsg	If this value is set to 1, this message will be marked as a moderator message.
MsgType	If this value is set to 1, this message will be marked as a sticky message.
Signature	If this value is set to 1, the user's signature will be stored with this message. Extended Forum option.
Title	The message title.

ForumNewMessageExtLogin.pbo

This file is rendered in preference to ForumNewMessage.pbo when extended login is enabled. ForumNewMessageExtLogin.pbo has the same variables as ForumNewMessage.pbo

ForumNewMessageUnavailable.pbo

This file is rendered if there is no write access to the database. ForumNewMessageUnavailable.pbo has the same variables as ForumNewMessage.pbo

pbs Captcha

<pbs:captcha>

<pbs:captcha> most run without cache. We recommend what you run this pbs tags in a Ajax call with a misc action. (misc is not cached with our FastCache level.)$('#Newsfeed_bottom').load("/apps/pbcs.dll/misc?url=/misc/captcha.pbs&omniture=0");

The <pbs:captcha> tag represents a visual captcha. Currently, the tag can be used everywhere, but server side validation is only implemented for forums and xsendmail for now.

The tag supports the following options:

Length	The number of characters to render in the captcha. Default is 6 characters.
BackgroundColor	Sets the color of the captcha background, use hex RGB value.
FontColor	Sets the color of the captcha font, use hex RGB value.
NoiseColor	Sets the color of the captcha noise, use hex RGB value.
ObjectClass	Specifies what object file design class to use.

If there are no color values, or only some values set on the tag, it looks for any corresponding values in the publicus.ini. If there are no values at all, it defaults to the original color scheme (white/blue/light blue). Faulty written color values return black as the default color.

```
<pbs:!captcha length=N>
```

```
<pbs:!captcha Length=5 FontColor=RRGGBB NoiseColor=RRGGBB BackgroundColor=RRGGBB
ObjectClass=200>
```

The tag outputs the content of its object file, Captcha.pbo, specifically the file Captcha200.pbo.

Check a user's solution

Checking if a solution is correct can be done in the pbcs.dll by using the following action:

```
<%pbcs.dll%>/checkcaptcha?CaptchaID=X&Captcha=Y&Unpersist=Z
```

|| CaptchaID | The <%ID%> in captcha object file. ||

Captcha	The user's solution to the captcha.
Unpersist	Whether or not to remove the captcha from the user's session file. Default is 0. The unpersist can be forced with the **ForceUnpersist** setting in the CAPTCHA section

The action returns an HTTP response of **200 OK** if the check is successful and **400 BAD REQUEST** if the check fails.

Captcha.pbo

Captcha.pbo supports the following variables:

<%ID%>	The ID of the captcha.
<%image%>	The image URL to the visual captcha.
<%length%>	The number of characters rendered.

An example object file (integrated into a form contructed as a simple name-value table:

```
[%{image}
  <tr>
    <td width="22%">
      <b>Captcha</b>
    </td>
    <td width="78%" valign="middle">
      <input type="hidden" name="CaptchaID" value="<%ID%>">
      <img src="<%Image%>" border="0" title="Captcha"> <input type="text" name="Captcha"
maxlength="<%Length#+5%>" size="<%Length#+3%>">
    </td>
  </tr>
%]
```

pbs forumoptions

<pbs:forumoptions>

The user preference function is used to set the preferences that apply to the Forum Module in SAXOTECH Online.

Template

Gen_forumoptions.pbs.

ForumPreferences.pbo

This object file brings up the current information, in addition to saving the data input by the user. The output fields are listed with their input fields.

Input fields

Avatar	Used to attach a picture to the user information. This picture can then be shown with every post the user makes.
	`<input type="file" name="Avatar">`
	`<img src="<%avatarlink%>&maxw=100&NoCache=1">`
	Delete
	`<input type="checkbox" name="DeleteAvatar" value="1">`

First name	Captures the first name for the user. `<input type="text" name="FirstName" value="<%FirstName%>">`
Form setup	The form must be set up so the posted information will be saved. `<form method="POST" Action= "<%pbcs.dll%>/forumoptions?Category=<%Category%>&SavePreferences">`
Last name	Captures the user's last name.<input type="text" name="LastName" value="<%LastName%>">
Nickname	Captures a nickname. The nickname is displayed instead of the user's real name in the forum listing. `<input type="text" name="NickName" value="<%NickName%>">`
Signature	Captures the signature. It will be attached to every posting. `<textarea name="Pref_Signature" rows=6 cols=45 ><%Signature%></textarea>`

Output fields

<%\Author%>	The currently logged in user's nickname. If there is no nickname, the first name followed by the last name.
<%AvatarLink%>	The currently logged in user's avatar image link.
<%Category%>	The template category that is used to resolve the correct pbo files in the object's directory.
<%\EMail%>	The currently logged in user's email address.
<%FirstName%>	The currently logged in user's first name.
<%IsModerator%>	This is set to 1 if the currently logged in user is a moderator.

<%LastName%>	The currently logged in user's last name.
<%Name%>	The currently logged in user's first name followed by the last name.
<%NickName%>	The currently logged in user's nickname.

ForumPreferencesSaved.pbo

<%\Author%>	The currently logged in user's nickname. If there is no nickname, then it is the first name followed by the last name.
<%AvatarLink%>	The currently logged in user's avatar image link.
<%Category%>	The template category that is used to resolve the correct pbo files in the objects directory.
<%\EMail%>	The currently logged in user's email address.
<%FirstName%>	The currently logged in user's first name.
<%IsModerator%>	This is set to 1 if the currently logged in user is a moderator.
<%LastName%>	The currently logged in user's last name.
<%Name%>	The currently logged in user's first name followed by the last name.
<%NickName%>	The currently logged in user's nickname.

ForumPrivateEmail.pbo

Used to allows users to e-mail each other when the user's email addresses are not shown on the forum page.

This is used in conjunction withForumPrivateEmailMessage.pbo and ForumPrivateEmailSent.pbo

URL

The URL for the form is generated in the message listing and should look something like this:

```
/apps/pbcs.dll/forumoptions?Category=forum
*&Privateemail=<%PostedBy%>*
```

Input fields

Form	The form must be set up like this: `<form method="POST" Action= "<%pbcs.dll%>/forumoptions?Category=<%Category%>&SendPrivateEmail" >`
To Address	The to address must be set up in a hidden field like this: `<input type="hidden" name="EMailTo" value="<%EMailTo%>">`
From Name	The sender's name. `<input type="hidden" name="FromName" value="<%Name%>">`

From Address	The sender's address. `<input type="hidden" name="FromAddress" value="<%\EMail%>">`
Email subject	Captures the email subject. `<input type="text" name="Subject" maxlength=255 size=60 class=forum>`
Message body	The message body. `<textarea name="MessageBody" rows=20 cols=45 class=forum></textarea>`

Output fields

<%\Author%>	The name of the user that created the message.
<%AvatarLink%>	The currently logged in user's avatar image link.
<%Category%>	The template category that is used to resolve the correct pbo files in the object's directory.
<%\EMail%>	The currently logged in user's e-mail address.
<%EmailTo%>	The email address of the user that created the message.
<%FirstName%>	The currently logged in user's first name.
<%IsModerator%>	This is set to 1 if the currently logged in user is a moderator.
<%LastName%>	The currently logged in user's last name.
<%Name%>	The currently logged in user's first name followed by the last name.
<%NickName%>	The currently logged in user's nickname.

ForumPrivateEmailMessage.pbo

Used to allows users to e-mail each other when the user's email addresses are not shown on the forum page.

This is used in conjunction with ForumPrivateEmail.pbo and ForumPrivateEmailSent.pbo

<%\Author%>	The currently logged in user's nickname. If there is no nickname, then it is the first name followed by the last name.
<%AvatarLink%>	The currently logged in user's avatar image link.
<%Category%>	The template category that is used to resolve the correct pbo files in the object's directory.
<%\EMail%>	The currently logged in user's email address.
<%FirstName%>	The currently logged in user's first name.
<%IsModerator%>	This is set to 1 if the currently logged in user is a moderator.
<%LastName%>	The currently logged in user's last name.
<%Name%>	The currently logged in user's first name followed by the last name.
<%NickName%>	The currently logged in user's nickname.

ForumPrivateEmailSent.pbo

Used to allows users to email each other when the user's email addresses are not shown on the forum page.

This is used in conjunction with ForumPrivateEmail.pbo and ForumPrivateEmailMessage.pbo.

<%\Author%>	The recipient name.
<%AvatarLink%>	The recipient's avatar image link.
<%Category%>	The template category that is used to resolve the correct pbo files in the object's directory.
<%\EMail%>	The email address of the recipient.
<%FirstName%>	The recipient's first name.
<%FromAddress%>	The email address of the currently logged in user.
<%FromName%>	The name of the currently logged in user.
<%IsModerator%>	This is set to 1 if the recipient is a moderator.
<%LastName%>	The recipient's last name.
<%MailSent%>	Has a value of 1 if the mail is sent successfully.
<%MessageBody%>	The message body of the private mail that was sent.
<%Name%>	The recipient's first name followed by the last name.
<%NickName%>	The recipient's nickname.
<%Subject%>	The message subject of the private mail that was sent.

pbs news_content

<pbs:news_content>

Used to show the list of themes on your site and to view messages.

Template

The default template for the forums is **gen_news.pbs**. You can also use this tag on other templates.

URL

The main URL for forums is **/apps/pbcs.dll/news_themes?Category=FORUM**.
The **Category** is optional, but should be specified so you can view statistics on forum traffic.

Options

All options are listed here, even if they are used on other pages than the theme listing.

Ascthreadsort	Makes the message threads sort in ascending order on the messages page. The default is to sort the message threads in descending order.
Conditional	Enables the listing of theme and article forums. Without this option, only group forums are shown.
Expandthreads	Enables the expansion of threads in the messages overview so you can see all stories in the thread you are viewing. This option cannot be used with a protected forum.
Hotthreshold	Sets the threshold for when a message thread is hot. The default threshold is 10 replies in 24 hours. Enter `hotthreshold=25` if the message must have 25 replies in 24 hours before it is considered hot.

Maxdays	Sets how many days back the tag should look for forum postings. By default, this option is not used.To set the maximum days the tag should look back for forum postings, enter
	`maxdays=X`
	where
	`X`
	is the number of days.

ShowHeaders	Enables message headers in the group listing. This option can only be used if you have very few messages. Showing headers on the group page can make your site fail. This option is turned off by default. Enter `showheaders=1` to enable this option.

Threadcount	Sets the maximum number of threads that should be shown on the overview page. You can use previous and next links to navigate to the other threads. By default, all threads are shown on one page. Enter
	`threadcount=X`
	for
	`X`
	threads to be shown per page. You must use the functions for previous and next links. Otherwise, web guests cannot navigate to all message threads.

ForumMessage.pbo

This function displays the actual message. The link to this page is automatically built by the system with the pbs news_content tag.

Template

The default template for forums is **gen_news.pbs**. The template is accessed through this URL:

```
/apps/pbcs.dll/news_message?ThemeID=1&GroupID=7&InReplyTo=16
```

Theme ID, Group ID and InReplyTo must be specified.

This is used in conjunction with ForumMessageAttachmentItem.pbo and ForumMessageThreadItem.pbo.

<%ArtLink%>	A link to the originating story if the forum is an Article Forum.
<%Attachments%>	Lists file attachments. Please see ForumMessageAttachmentItem.pbo
<%\Author%>	The name of the person who wrote the message. The user's nickname is shown if the user has registered a nickname. Otherwise, the name with which the user registered is shown.
<%AvatarLink%>	The link to the image associated with the user.
<%Body%>	The message body.
<%Created%>	The date/time the message was written.

<%DeleteMessageLink%>	A link to an action that will cause the top message to be deleted with all its sub messages. Deleting messages actually only marks those messages as unpublished.
<%DeleteReplyLink%>	A link to an action that will cause the current message to be deleted with all its sub messages. Deleting messages actually only marks those messages as unpublished.
<%Description%>	The name/description of the group in which the message is posted.
<%EditMessageLink%>	The link to the page for editing the message. This feature is only available to the original poster.
<%\Email%>	The email address of the author.
<%GroupLink%>	The link back to the group listing.
<%GroupOptions%>	This field contains an HTML option fragment that contains a list of group names and IDs for the theme in which the current message exists. This is typically used to present a drop-down list of groups that the user can use to navigate to a new group.
<%HasNew%>	This has a value of 1 when the current message has replies that have been added since the current user's last visit date/time.
<%HotThread%>	This field has the value 1 if the thread is hot. The **Hotthreshold** parameter on the <pbs:news_content> tag controls the number of replies required before a message is considered hot.
<%Images%>	Lists image attachments.
<%IsGrpLocked%>	This field has a value of 1 when the group has been locked.
<%IsModerated%>	The group in which this message exists is moderated.
<%IsMsgLocked%>	This has a value of 1 if the message is locked.
<%IsNew%>	This has a value of 1 when the current message has been created since the current user's last visit date/time.
<%IsoCreated%>	The date/time the message was written, in ISO format
<%ISOLastMessageDate%>	The same as `<%LastMessageDate%>` except in ISO format.
<%IsoPublished%>	The date/time the message was published, in ISO format. This value is used when the forum is moderated. If moderation is not enabled, creation date/time and published date/time will be the same.
<%ISOUserRegDate%>	The date the user was added to the user database.
<%IsSticky%>	This has a value of 1 if the message is a sticky message. A sticky message will appear above all normal messages.
<%LastMessageDate%>	The date/time the last message in the thread was created.
<%LastPoster%>	The nickname of the user that last posted to this thread.
<%Link%>	The link to the full text message.
<%MessageID%>	The ID for the message.
<%MessageThreads%>	Lists replies to this message in a threaded style forum. By including this variable, you define your forum as a threaded forum.
<%NewMessageLink%>	The link to the page for posting a new top level message.
<%PostedBy%>	The numeric User ID.
<%PostedByHost%>	The IP address of the user who posed the message.
<%Published%>	The date/time the message was published. This value is used when the forum is moderated. If moderation is not enabled, creation date/time and published date/time will be the same.
<%Replies%>	Lists replies to this message in a non-threaded style forum. By including this variable you define your forum as a non-threaded forum.
<%ReplyMessageLink%>	The link to the page for posting a reply.
<%ShowRepliesLink%>	The same as `<%Link%>`

<%Signature%>	The user's signature.
<%Theme%>	The name of the theme in which the message is posted.
<%ThemeDescription%>	The theme description for this message.
<%ThemeLink%>	A link to the list of groups.
<%ThreadCount%>	The number of replies to this message.
<%\Title%>	The title of the message.
<%Today%>	This has a value of 1 when there is a reply to the current message that has been created today.
<%ToggleLockLink%>	A link that will cause the messages locked status to toggle. A locked message will become unlocked and an unlocked message will become locked.
<%UserLevel%>	This field contains the user level.
<%UserPostCount%>	The total number of posts by the user.
<%LoggedInUserVerified%>	This field has a value of 1 if the currently logged on user is verified.
<%LoggedInUserLevel%>	The user level of the currently logged on user.

ForumMessageAttachmentItem.pbo

This file is used to list all attachments for a given forum message. It is used in conjunction with ForumMessage.pbo and ForumMessageThreadItem.pbo.

<%Counter%>	The counter for the attachments.
<%DownloadLink%>	The link that will download the attachment.
<%Filename%>	The file name of the attachment.
<%ImgLink%>	The link that will display the attachment, if it is an image.
<%IsImage%>	This field has a value of 1 if the attachment is an image.
<%IsLastItem%>	This field has a value of 1 if the attachment is the last in the list.

ForumMessageThreadItem.pbo

This file is used to list all messages in the thread. It is used in conjunction with ForumMessageAttachmentItem.pbo and ForumMessageThreadItem.pbo.

It is also used on the group overview page. Please see ForumMessagesMessageThreadItem.pbo. The field

```
<%MessageThreads%>
```

must be present in the ForumMessages.pbo for this field to be used.

<%Attachments%>	Lists file attachments. Please see ForumMessageAttachmentItem.pbo.
<%AttachmentsNotImage%>	The number of non image attachments associated with this message. This is the same as <%Attachments%> minus <%Images%>.
<%\Author%>	The name of the person who wrote the message.
<%Body%>	The body text of the current message.
<%Category%>	The template category that is used to resolve the correct pbo files in the object's directory.
<%DeleteMessageLink%>	A link to an action that will cause the top message to be deleted with all its sub messages. Deleting messages actually only marks those messages as unpublished.
<%DeleteReplyLink%>	A link to an action that will cause the current message to be deleted with all its sub messages. Deleting messages actually only marks those messages as unpublished.
<%EditMessageLink%>	The link to the page for editing the current message.
<%\Email%>	The email address of the message author.

<%GroupLink%>	A link to the list of messages for a link.
<%HasNew%>	This has a value of 1 when the current message has replies that have been added since the current user's last visit date/time.
<%HotThread%>	This field has the value 1 if the thread is hot. The **Hotthreshold** parameter on the `<pbs:news_content>` tag controls the number of replies required before a message is considered hot.
<%Images%>	(Threaded Forums only) The number of image attachments associated with this message.
<%Indent%>	(Threaded Forums only) This field is used to place message details at the appropriate position on the screen, depending on the level of threading.
<%IsGrpLocked%>	This has a value of 1 if the group in which this message belongs is locked.
<%IsMsgLocked%>	This has a value of 1 if the message is locked.
<%IsNew%>	This has a value of 1 if the current message has been created since the current user's last visit date/time.
<%ISOCreated%>	The same as `<%Created%` except in ISO format.
<%ISOLastMessageDate%>	The same as `<%LastMessageDate%>` except in ISO format.
<%ISOPublished%>	The same `<%Published%>` except in ISO format.
<%IsSticky%>	This has a value of 1 if the message is a sticky message. A sticky message will appear above all normal messages.
<%LastMessageDate%>	The date/time the last message in the thread was created.
<%Link%>	The link to the full text message.
<%MessageID%>	A value the represents the current message as a number.
<%NewMessageLink%>	The link to the page for posting a new top level message.
<%Published%>	For moderated forums, this is the date/time the moderator published the message. For non-moderated forums, this is the date/time the message was created.
<%ReplyMessageLink%>	The link to the page for posting a reply.
<%ShowRepliesLink%>	The same as `<%Link%>`
<%ThemeLink%>	A link to the list of groups for the theme.
<%ThreadCount%>	The number of replies to this message.
<%\Title%>	The title of the message.
<%Today%>	This has a value of 1 when there is a reply to the current message that has been created today.
<%ToggleLockLink%>	A link that will cause the messages locked status to toggle. A locked message will become unlocked and an unlocked message will become locked.
<%TopThreadMessageID%>	The message ID of the original posting.
<%TreeLines%>	(Threaded Forums only) This lists tree-line characters appropriate to the level of indent for this thread item.

ForumThemeItem.pbo

Used to build each item in the list.

<%Counter%>	The counter for the theme items.
<%Description%>	The description of the theme.
<%GroupItems%>	Outputs the group items if `showgroups=1` is on the URL.

<%GroupsListLink%>	The link to the theme listing.
<%HasNew%>	This field is set to 1 if there are messages or replies created after the reader last view this page. This option uses a cookie.
<%IsLastItem%>	This field contains the value 1 if it is the last theme item listed.
<%ISOLastMessageDate%>	The last message date in ISO format.
<%LastMessageDate%>	The most recent date on which a message was posted in the theme.
<%LogoLink%>	The link for the logo, if the theme has a logo
<%MessageCount%>	The number of messages in the theme, plus all the replies.
<%Name%>	The name of the theme.
<%Theme%>	The ID for the theme.
<%ThemeLogoLink%>	This is the same as `<%LogoLink%>`.
<%Today%>	This field has the value 1 if the theme has a new message today.
<%TopMessageCount%>	The number of messages in the theme.

ForumThemes.pbo

Used to build the main theme listing.

<%ISOUserLastVisitedDate%>	Outputs the date of the last visit for this reader.
<%ThemeItems%>	Outputs the themes.

ForumThemesGroupItem.pbo

This file is used to list groups with each theme. Theme groups can be listed by entering

```
ShowGroups=1
```

on the URL, and by defining ForumThemesGroupItem.pbo.

ForumThemesGroupItem.pbo exposes the same variables as ForumGroupItem.pbo.

pbs news_count

<pbs:news_count>

This tag is used to show the number of messages in the forums. The tag will show the total number of messages on the level on which the tag is used. If the tag is used on the main overview page, it will show the total number of messages. If it is used on the group page, it will show the total number of messages for the group.

Template

This tag can be used on all templates.

Options

Group	Enables you to specify the group for which the tag should show the total number of messages. The theme must also be specified if this option is used. By default, all groups are counted.You can use pbs queryparam to set this value, or you can hardcode it. The group ID is used.
Theme	Enables you to specify the theme for which the tag should show the total number of messages. By default, all themes are counted.You can use pbs queryparam to set this value, or you can hardcode it. The theme ID is used.

pbs news_grouplink

<pbs:news_grouplink>

This tag is used to link to a forum from a story.

Options

Articleforum	Makes the system link to the article forum for the story.
Autoforum	Makes the system link to the correct forum type based on the forum setting in the story. You should use this option if you want to link from the story to the forum.
Conditional	Used with Themeforum, Articleforum and Groupforum to make these options only output a link if forum is enabled in the story.
Groupforum	Makes the system link to the group forum for the story.
Themeforum	Makes the system link to the theme forum for the story.

ForumGroupLink.pbo

<%Link%>	The link to the forum.

pbs news_groups

<pbs:news_groups>

The group listing lists all groups in a theme. The link to this page is automatically built by the system by the pbs news_content tag on the theme page.

This tag can be used on all pages. The theme number is used and must be specified when you use this tag. This tag can also be used to list a single newsgroup.

Template

The default template for forums is called **gen_news.pbs**. It is accessed through this URL:

```
/apps/pbcs.dll/news_groups?Category=FORUM&ThemeID=2
```

Theme ID must be specified. Category is optional, but should be specified so you can view statistics for the forum.

Options

Theme	Specifies the theme for which you want to list groups.

This tag also uses the same options as pbsnews_content.

ForumGroupHeadlineItem.pbo

Used to show the headlines belonging to a group directly in the group listing. If you have a lot of messages, this file will make your forum pages very slow.

<%Attachments%>	A value for the number of file attachments that are associated with this post.
<%AttachmentsNotImage%>	The number of non image attachments associated with this message. This is the same is <%Attachments%> minus <%Images%>
<%\Author%>	The name of the person who wrote the message.
<%Counter%>	This is an index value for the headline item. The count starts with a value of 1.
<%\Email%>	The email address of the message author.
<%HasNew%>	This field is set to 1 if the message was created after the reader viewed the page. This option uses a cookie.
<%Images%>	The number of image attachments associated with this message.

<%IsLastItem%>	This field contains the value 1 if it is the last field listed.
<%LastThreadDate%>	The publication date for the newest message in the thread.
<%Link%>	The link to the message.
<%ThreadCount%>	The number of messages in the thread. This field cannot be used if the option Expandthreads is used in the tag.
<%\Title%>	The title of the message.

ForumGroupItem.pbo

Used to build the item list.

<%AllMessageListLink%>	A link to the story listing that shows stories according to the FORUM section#MaxDaysInactive setting in the Publicus.ini.
<%Counter%>	The counter for the groups.
<%Description%>	The description of the group.
<%GroupLogoLink%>	The link for the logo, if the group has a logo.
<%HasNew%>	This field is set to 1 if there are messages or replies created after the reader last viewed the page. This option uses a cookie.
<%Headlines%>	This field can be used to show the threads in a group. This option requires that the **ShowHeaders** option is enabled. This option should not be used.
<%ISOLastMessageDate%>	The same as **LastMessageDate**, but in ISO format.
<%ISOUserLastVisitedDate%>	The date this reader last visited the page.
<%IsGrpLocked%>	This field is set to 1 if the group has been locked.
<%IsLastItem%>	This field contains the value 1 if it is the last field listed.
<%LastMessageDate%>	The date of the last posted message.
<%LogoLink%>	The link for the logo, if the group has a logo.
<%MessageCount%>	The number of messages posted.
<%MessageListLink%>	A link to the story listing.
<%Moderator%>	The name of the moderator
<%Name%>	The name of the group.
<%NewMessageLink%>	A link to the new message page.
<%TopMessageCount%>	The number of messages in the group.

ForumGroups.pbo

Shows the groups belonging to a theme.

<%GroupItems%>	Lists all the group items.
<%ISOUserLastVisitedDate%>	Contains the date of the last visit for this reader.
<%ShowAllLink%>	A link showing all messages, regardless if they are older than the number of days set in the ini file for deactivation of threads. It is used like this: `<a href="<%showalllink%>">Show all messages including inactive</a>` This option can only be used if the option ShowHeaders is used in the tag.
<%Theme%>	The name of the theme being viewed.
<%ThemeDescription%>	The description of the parent theme.

<%ThemeID%>	The ID of the parent theme.
<%ThemeListLink%>	The link to the main listing of themes.
<%ThemeLogoLink%>	A link to the themes logo image, if it has one.

pbs news_hotthreads

<pbs:news_hotthreads>

This tag is used to list messages that have the most activity.

Options

Category	Sets the category to be used when showing the message. This is done to show the forum messages in a category other than the category on which the messages is listed. By default this is not set. Enter `category=categoryid` to show the message using the template corresponding to the category ID.
Count	Controls how many messages the tag will show. The default is 10.The tag can show from 1 to 99 messages.
Group	Limits the groups from which the tag should show hot messages. The default is to use all groups. Enter `group=groupnumber` The group number can be found by looking at the links in the group listing for a theme.

Maxdays	Sets how many days back the tag should look for forum postings. By default, this option is not used.To set the maximum days the tag should look back for forum postings, enter
	`maxdays=X`
	where
	`X`
	is the number of days.

ObjectClass	Sets the object class. By default this is not used. Enter
	`objectclass=X`
	to use class
	`X`
	is used. Legal values are 1 through 99.

Theme	Limits the themes from which the tag should show hot messages. The default is to show all themes when viewing any page that is not showing a theme or a category. When viewing a theme or a category, the tag will only list new messages in the theme or category. Enter
	`theme=themenumber`
	The theme number can be found by looking at the links in the main forum listing. If
	`theme=0`
	is used, all messages will be shown on all pages.

ForumHotThreadItem.pbo

<%Attachments%>	A value for the number of file attachments that are associated with this post.
<%AttachmentsNotImage%>	The number of non image attachments associated with this message. This is the same as <%Attachments%> minus <%Images%>.
<%\Author%>	The author of the message.
<%Counter%>	The message counter for the list. This field is increased by one for each message.
<%Created%>	The date the message was created.
<%\Email%>	The e-mail address of the message author.
<%Group%>	The group under which the message is posted.
<%GroupLink%>	The link to the group under which the message is posted.
<%Images%>	The number of image attachments associated with this message.
<%IsLastItem%>	This field lists all entries.
<%Link%>	The link to the message.
<%MessageId%>	A value that represents the current message as a number.
<%MessageText%>	The message text.
<%Replies%>	The number of replies to the message.
<%Theme%>	This field contains the theme under which the message is posted.
<%ThemeLink%>	This field contains the link to the theme under which the message is posted.

ForumHotThreads.pbo

<%items%>	This field lists all entries.

pbs news_latest

<pbs:news_latest>

This function lists the newest messages in the forums.

Best practice:

Is to add this tag in a macro, because then the same cache file will be reused, instead of it always generates a new one, example:
template: <pbs:!macro name="getnewslatest" CacheTime="15">
\macro\getnewslatest.pbo: <pbs:news_latest count=5 showref="0">

Options

Category	Sets the category to be used when showing the message. This is done to show the forum messages in a category other than the category on which the messages are listed. By default the category is not set. Enter `category=categoryid` to show the message with the template associated with the category ID.
Count	Controls how many messages the tag will show. The default is 10. Enter `count=X` to show X messages. Legal values are 1 to 99.
Group	Limits the groups from which the tag should show newest messages. The default is to show newest messages from all groups. To use this option, set `group=groupnumber` The Group number can be found by looking at the links in the group listing for a theme.
ObjectClass	This option sets the object class. By default the object class is not set. Enter `objectclass=X` to use class X.Legal values are 1 through 99.
Showref	Controls how many characters of the message should be shown by the tag. The default is 0.The tag can be set from 0 to 999.

Theme	Used to list the newest messages for the specified theme only. By default, this setting is not used. Enter
	`theme=themenumber`
	to use this option. The theme number can be found by looking at the links in the main forum listing. Example:
	`<pbs:news_latest Category=phpforums_rss Count=3 showref=120 theme=1201>`
	To list the newest messages for the entire forum, do not use the theme parameter. Example:
	`<pbs:news_latest Category=phpforums_rss Count=3 showref=120>`

ForumLatestMessageItem.pbo

<%Attachments%>	Contains a value for the number of file attachments that are associated with this post.
<%AttachmentsNotImage%>	The number of non image attachments associated with this message. The same as `<%Attachments%>` minus `<%Images%>`
<%Author%>	The author of the message.
<%Counter%>	The message counter for the list. This field is increased by one for each message.
<%Created%>	The date the message was created.
<%Description%>	The description of the group.
<%Email%>	The email address of the message author.
<%Group%>	The group under which the message is posted.
<%GroupLink%>	The link to the group under which the message is posted.

<%Images%>	The number of image attachments associated with this message.
<%IsLastItem%>	This field contains the value 1 if it is the last theme item listed.
<%IsNew%>	This field is set to 1 if the message was created after the reader last viewed the page.This option uses a cookie.
<%Link%>	The link to the message.
<%LinkTop%>	The link to the top message.
<%MessageId%>	A value that represents the current message as a number.
<%MessageText%>	The message text. The amount of text is limited by the **showref** option in the tag.
<%Theme%>	The theme under which the message is posted.
<%ThemeDescription%>	The description of the theme.
<%ThemeLink%>	The link to the theme under which the message is posted.
<%Title%>	The title of the message.
<%Updated%>	The date the thread was last updated.

ForumLatestMessages.pbo

<%items%>	This field lists all entries.

pbs news_messages

<pbs:news_messages>

The message listing lists all messages in the group. The link to this page is automatically built by the the system by the pbs news_content tag on the groups page.

The tag is is used like this:

```
<pbs:news_messages articleforum objectclass=2 conditional>
```

You can also list forum messages on a story page if the forum is attached to a category or a story.
`<pbs:news_messages>` should be used uncached

```
<pbs:!news_messages>
```

when it is used on article templates.

Template

The default template for forums is **gen_news.pbs**. It is accessed through this URL:

```
/apps/pbcs.dll/news_messages?ThemeID=2&GroupID=3&Category=FORUM
```

Theme ID and group ID must be specified. Category is optional.

Options

Articleforum	Used to enable the Article Forum. It must be present when you use the tag on a story template.
photogalleryforum From version 6.4.0	Used to enable the forum in photo galleries. It must be present when you use the tag on a photogallery template.
cceforum From version 6.4.0	Used to enable the forum in CCE. It must be present when you use the tag on a CCE template.

Autoforum	Lists all the messages in the forum type based on the forum setting in the story.
Class	Sets the object class.
Conditional	Makes the forum only appear on stories where forum is enabled.
Groupforum	Lists all the messages in the group forum for the story.
Themeforum	Lists all the messages in the theme forum for the story.
Threadcount	Sets the maximum number of threads to be shown. You can use previous and next links to get to the other threads. By default, all threads are shown on one page.

ForumMessageItem.pbo

This object file is used for every top-level message.

Please note only the Extended Forum can use all the fields. The Extended Forum is set up in the Publicus.ini file. For more information, please see VerifyUsers. If Extended Forum is **not** enabled, the following fields will not be available:

```
<%AvatarLink%>
<%ISOUserRegDate%>
<%PostedBy%>
<%PostedByHost%>
<%Signature%>
<%UserLevel%>
<%UserPostCount%>
<%UserRegDate%>
```

<%Attachments%>	This field contains a value for the number of file attachments that are associated with this post.
<%AttachmentsNotImage%>	The number of non-image attachments associated with this message. This is the same is <%Attachments%> minus <%Images%>.
<%\Author%>	The name of the person that wrote the message.
<%AvatarLink%>	A link to the author's avatar image. This is blank if the author does not have an avatar image.
<%Body%>	Outputs the body of the message.
<%Created%>	The date/time the message was written.
<%DeleteMessageLink%>	A link to an action that will cause the top message to be deleted with all its sub messages. Deleting messages actually only marks those messages as unpublished.
<%Description%>	The group name for this message.
<%EditMessageLink%>	The link to the page for editing the current message.
<%\Email%>	The email address of the author.
<%GroupLink%>	Link to the list of messages for a link.
<%GroupOptions%>	This field contains an HTML option fragment that contains a list of group names and IDs for the theme in which the current message exists. This is typically used to present a drop-down list of groups that the web guest can use to navigate to a new group.
<%HotThread%>	This field has the value 1 if the thread is hot.
<%Images%>	The number of image attachments associated with this message.
<%IsGrpLocked%>	This has a value of 1 if the group in which this message belongs is locked.
<%IsModerated%>	The group in which this message exists is moderated.
<%IsMsgLocked%>	This has a value of 1 if the message is locked.
<%IsoCreated%>	The date/time the message was written, in ISO format.
<%IsoPublished%>	The date/time the message was published, in ISO format. This value is used when the forum is moderated. If moderation is not enabled, creation date/time and published date/time will be the same.
<%ISOUserRegDate%>	The same as <%UserRegDate%> but in ISO format.
<%IsSticky%>	This has a value of 1 if the message is a sticky message. A sticky message will appear above all normal messages.

<%LastMessageDate%>	Outputs the date of the newest reply to the message.
<%Link%>	The link to the full text of the message.
<%MessageID%>	A value that represents the current message as a number.
<%MessageThreads%>	The listing of replies to the message. This field cannot be used with the Protected Forum option.
<%NewMessageLink%>	The link to the page for posting a new top level message.
<%NoVoteCount%> **From version 6.4.0**	Number of users who have voted no.
<%NoVoteLink%> **From version 6.4.0**	No vote link. The user must be logged in before voting
<%PostedBy%>	The numeric user ID.
<%PostedByHost%>	The IP address of the user who posed the message.
<%Published%>	The date/time the message was published. This value is used when the forum is moderated. If moderation is not enabled, creation date/time and published date/time will be the same.
<%ShowAllLink%>	A link to a list of messages for this message group that is not restricted by **MaxDays**.
<%ShowRepliesLink%>	The same as <%Link%>
<%Signature%>	The signature text of the message author.
<%Theme%>	The theme name for this message.
<%ThemeDescription%>	The theme description for this message.
<%ThemeLink%>	A link to the list of groups.
<%\Title%>	The title of the message.
<%ToggleLockLink%>	A link that will cause the messages locked status to toggle. A locked message will become unlocked and an unlocked message will become locked.
<%UserId%> **From version 6.4.0**	The id of the user.
<%UserLevel%>	The user level of the message author.
<%UserPostCount%>	The total number of posts the author has made in the system.
<%UserRanking%> **From version 6.4.0**	The ranking given by the specific user on an object.
<%UserRegDate%>	The date the user first registered in the system.
<%YesVoteCount%> **From version 6.4.0**	Number of users who have voted yes.
<%YesVoteLink%> **From version 6.4.0**	Yes vote link. The user must be logged in before voting.

ForumMessages.pbo

This file builds the framework for the message list.

<%ArtLink%>	A link back to the story, if the forum is an Article Forum.
<%Description%>	The description of the group.
<%GroupLogoLink%>	The same as <%LogoLink%>.
<%GroupsListLink%>	The same as <%ThemeLink%>.
<%IsGrpLocked%>	This field is set to 1 if the group is locked.
<%IsModerated%>	This field is set to 1 if the group is moderated.
<%LoggedInUserLevel%>	The user level of the user currently logged on to the forum.
<%LoggedInUserVerified%>	This field is set to 1 if the user currently logged on is verified.

<%LogoLink%>	The link for the logo, if the group has a logo.
<%Messageitems%>	This parameter lists all the top level messages in the group.
<%NavBar%>	Lists a page navigation area. This functionality provides a more powerful alternative to using NextLink, PageNum, NumPages and PrevLink.
<%NewMessageLink%>	The link to the new message page.
<%NextLink%>	The link to the next page. This option is used when you use **Threadcount** to limit the number of threads on one page.
<%NumPages%>	The number of pages that are required to display all the messages. The option is used when you use **Threadcount** to limit the number of threads on one page.
<%PageNum%>	When messages are displayed across several pages, this field contains a numeric value for the page index currently displayed. The option is used when you use **Threadcount** to limit the number of threads on one page.
<%PrevLink%>	The link to the previous page. This option is used when you use **Threadcount** to limit the number of threads on one page.
<%ShowAllLink%>	A link showing all messages, including those older than the cutoff date in the Publicus.ini. The link is used like this: `<a href="<%showalllink%>">Show all messages including inactive</a>`
<%Subject%>	The description of the group.
<%Theme%>	The name of the theme.
<%ThemeDescription%>	The description of the theme.
<%ThemeLink%>	The link back to the group listing.
<%ThemeLogoLink%>	The link for the theme logo, if the theme has a logo.

ForumMessagesMessageThreadItem.pbo

This object file is used to list the replies to a top level message. If the file does not exist, ForumMessageThreadItem.pbo will be used.

Please note only the Extended Forum can use all the fields. The Extended Forum is set up in the Publicus.ini file. For more information, please see VerifyUsers. If the Extended Forum is not enabled, the following fields will **not** be available:

```
<%AvatarLink%>
<%IsAdminUser%>
<%IsModerator%>
<%IsNormalUser%>
<%ISOUserRegDate%>
<%IsROnlyUser%>
<%LastPoster%>
<%PostedBy%>
<%PostedByHost%>
<%Signature%>
<%UserID%>
<%UserLevel%>
<%UserPostCount%>
<%UserRegDate%>
```

<%Attachments%>	A value for the number of file attachments that are associated with this post.
<%AttachmentsNotImage%>	The number of non-image attachments associated with this message. This is the same is <%Attachments%> minus <%Images%>.
<%\Author%>	The name of the person who wrote the message.
<%AvatarLink%>	A link to the author's avatar image. This is blank if the author does not have an avatar image.
<%Body%>	The body of the message.
<%Created%>	The date/time the message was written.

<%DeleteMessageLink%>	A link to an action that will cause the top message to be deleted with all its sub messages. Deleting messages actually only marks those messages as unpublished.
<%DeleteReplyLink%>	A link to an action that will cause the current message to be deleted with all its sub messages. Deleting messages actually only marks those messages as unpublished.
<%Description%>	The group description of this message.
<%EditMessageLink%>	The link to the page for editing the current message.
<%\Email%>	The email address of the author.
<%GroupLink%>	A link to the list of messages for a group.
<%GroupOptions%>	This field contains an HTML option fragment that contains a list of group names and IDs for for the theme in which the current message exists. This is typically used to present a drop-down list of groups, that the user can use to navigate to a new group.
<%HasNew%>	This has a value of 1 when the current message has replies that have been added since the current user's last visit date/time.
<%HotThread%>	This field has the value 1 if the thread is hot. Note the **Hotthreshold** parameter on the pbs news_content tag controls the number of replies required before a message is considered hot.
<%Images%>	The number of image attachments associated with this message.
<%Indent%>	Controls the story indent.
<%IsAdminUser%>	This value is 1 if the author of the message has a user level of 9.
<%IsGrpLocked%>	This field has a value of 1 when the group has been locked.
<%IsModerated%>	The group in which this message exists is moderated.
<%IsModerator%>	This value is 1 author of the message has a user level of 8.
<%IsMsgLocked%>	This has a value of 1 if the message is locked.
<%IsNew%>	This has a value of 1 when the current message has been created since the current user's last visit date/time.
<%IsNormalUser%>	This value is 1 if the author of the message has a user level in the range of 5 to 7.
<%ISOCreated%>	Same as <%Created%> except in ISO format.
<%ISOLastMessageDate%>	The same as <%LastMessageDate%> except in ISO format.
<%ISOPublished%>	The same as <%Published%> except in ISO format.
<%ISOUserRegDate%>	The same as <%UserRegDate%> but in ISO format.
<%IsROnlyUser%>	This value is 1 if the author of the message has a user level of 1.
<%IsSticky%>	This has a value of 1 if the message is a sticky message. A sticky message will appear above all normal messages.
<%LastMessageDate%>	The date/time the last message in the thread was created.
<%LastPoster%>	The nickname of the user that last posted to this thread.
<%Link%>	The link to the full text.
<%LoggedInUserLevel%>	The level of the user currently accessing the system.
<%LoggedInUserVerified%>	This field is set to 1 if the user currently accessing the system has been verified.
<%MessageID%>	A value representing the current message as a number.
<%NewMessageLink%>	The link to the page for posting a new top level message.
<%PostedBy%>	The numeric user ID.
<%PostedByHost%>	The IP address of the user who posed the message.
<%Published%>	For moderated forums, this is the date/time the moderator published the message. For non-moderated forums this is the date/time the message was created.
<%ReplyMessageLink%>	The link to the page for posting a reply.
<%ShowAllLink%>	A link to a list of messages for this message's group that is not restricted by **MaxDays**.

<%ShowRepliesLink%>	The same as <%Link%>
<%Signature%>	The signature text of the message author.
<%Subject%>	The group name for this message.
<%Theme%>	Theme name for this message.
<%ThemeDescription%>	The theme description for this message.
<%ThemeLink%>	Link to the list of groups.
<%ThreadCount%>	The number of replies to this message.
<%\Title%>	The title of the reply.
<%Today%>	This has a value of 1 when there is a reply to the current message that has been created today.
<%ToggleLockLink%>	A link that will cause the message's locked status to toggle. A locked message will become unlocked and an unlocked message will become locked.
<%TreeLines%>	Lists tree-line characters appropriate to the level of indent for this thread item.
<%UserID%>	An integer ID that represents the author in the system.
<%UserLevel%>	The user level of the author that created the message.
<%UserPostCount%>	The total number of posts the author has made in the system.
<%UserRegDate%>	The date the user first registered in the system.

ForumNavBar.pbo

This object file is used to list a navigation bar, which can help the user quickly navigate through many pages of message data.

<%FirstLink%>	The link to the first page, if it is not currently being viewed.
<%LastLink%>	The link to the last page, if it is not currently being viewed.
<%NavbarItems%>	This field lists all entries.
<%NextLink%>	The link to the next page, if there is a next page.
<%NumPages%>	The number of pages that are required to display all the messages. The option is used when you use **Threadcount** to limit the number of threads on one page.
<%PageNum%>	When messages are displayed across several pages, this field contains a numeric value for the page index currently displayed. The option is used when you use **Threadcount** to limit the number of threads on one page.
<%PrevLink%>	The link to the previous page, if there is a previous page.

ForumNavBarItem.pbo

This object file is used to list a navigation bar, which can help the user quickly navigate through many pages of message data.

<%CurrentPage%>	The page number that is displayed for the list of pages.
<%IsCurrent%>	This has a value of 1 if this item represents the current page in the list.
<%PageLink%>	A link to the message represented by this item.
<%\PageNumber%>	The page number of this item.

pbs search

<pbs:search>

You can search all messages in the forums.

The example code is set up so that you search within the scope you are viewing. For example, if you are viewing a group, you will only search within that group.

Template

This function can be placed on all templates.

Options

Encoding	Because the search URL is built using a PBS tag, the output cannot be encoded to ensure compliance with W3C standards. The
	<pbs:search> tag now accepts a parameter
	`encode="1"`
	that will encode the returned URL in the same way as using
	`$h`
	on a variable. For example:
	`<pbs:search encode="1">`
	will turn an ampersand character "&" into "&"

Search form

The form action must be {{<pbs:search type=forum>}}like this:

```
<form action="<pbs:search type=forum>" method="post">
```

Fields and buttons

Crit	Used to input the search text string.
Group	(Hidden field) Used to limit the search to the group being viewed or to a specified group. Leaving the field empty will make the form search all groups. Example: `<input type="hidden" name="group" value="<pbs:queryparam param=groupid>">` The tag pbs queryparam is used to output the group ID for the group.
Submit	Used to add a submit button.
Theme	(Hidden field) Used to limit the search to the theme being viewed or to a specified theme. Leaving the field empty will make the form search all themes. Example: `<input type="hidden" name="theme" value="<pbs:queryparam param=themeid>">` The tag pbs queryparam is used to output the theme ID for the theme.

URL parameters

SearchType	Used to qualify the search type. For example: `<SELECT NAME="SearchType" SIZE="1" onchange="SearchForm.action='<pbs:search type=forum>&SearchType=' + this.selectedIndex"><OPTION VALUE="*" >All Fields<OPTION VALUE="Body">Body<OPTION VALUE="Author">Author<OPTION VALUE="Title">Title<OPTION VALUE="EMail">EMail</SELECT>` It can have one of the following integer values: • 1 = Search all fields • 2= Search only the body field • 3= Search only the author field • 4= Search only the title field • 5= Search only the e-mail field

pbs searchresult and forums

<pbs:searchresult>

Template

This function uses the template **gen_forumsearch.pbs**.

Options

Count	Sets the number of messages that will be shown on the search result page.
Showref	Enter `showref=1` to enable the output of the body for the message.
Sites	Used to search multiple site codes. Usage: `sites=AA,BB,CC`

SearchResultForum.pbo

This file works in the same way as SearchResult.pbo

SearchResultForumItem.pbo

<%Body%>	The entire message body.
<%Counter%>	Countains the counter value.
<%Description%>	The description of the theme.
<%GroupLink%>	The link to the overview for the group.
<%GroupName%>	The name of the group to which the message belongs.

<%LeadIn%>	The first 250 characters of the message.
<%MessageLink%>	The link to the message.
<%PublishedDate%>	The published date for the message.
<%Subject%>	The description of the group.
<%ThemeLink%>	The link to the overview for the theme.
<%ThemeName%>	The name of the theme to which the message belongs.
<%\Title%>	This field contains the title of the message.

pbs userverify

<pbs:userverify>

When you use user registration, an e-mail must be sent to the user so she can verify that her e-mail address is correct. The system will automatically redirect the user to this page before she can post any messages.

Template

Gen_verify.pbs

UserVerificationMail.pbo

<%Accept%>	The link to the acceptance page.
<%ClientHost%>	The host name for the machine that triggered the verification process.
<%Decline%>	A link to decline the verification.
<%Name%>	The name of the person being verified.

Example

```
<P>Hi <%Name%>.
<P>Please click on the link immediately below this paragraph in order to
activate your new forum account on DailyNews.com:
<%Accept%>
<P>If you do not want to activate your account, or if you never requested the account,
please click on the following link:
<%Decline%>
<P>The request to create this account was made from <%ClientHost%>. If you
suspect that someone may have abused your account, please contact
<A HREF="mailto:john.smith@dailynews.com">john.smith@dailynews.com</A>.
```

UserVerify.pbo

<%EMailAddress%>	The e-mail address that is being verified.
<%Error%>	An error message if anything went wrong.
<%NextLink%>	A link back to the page that triggered the verification.
<%SendMailError%>	An error message if anything when wrong during the sending of the email.
<%Status%>	The status for the verification. The field is used to output different texts. Please look at the example to see what the different statuses mean.

Example

```
[%{Status=3}
A verification e-mail has been sent to <%EMailAddress%>. In order to use our forums, we need to
verify your email address. When you receive this e-mail, click on the link provided to verify
this account. Once you have verified your account, you can post in the forums.<br>
You should receive the verification e-mail within the next 5-10 minutes.<br>
%]
[%{Status=1}
Your user account has been verified. You can now proceed to post in our forums. If you want to
modify your user preferences, you can click
<a href="javascript:NewWindow(400, 400, '<%pbcs.dll%>/Forumoptions?Category=FORUMS');"
>here</a>.<br>
[%If you want to return to the location that triggered the verification process, click <a href=
"<%NextLink%>">here</a>.%]<br>
%]
[%{Status=0}
A verification email has already been sent. If you wish to resend the email, click <a href=
"userverify?Resend=1">here</a>.
%]
[%{Status=-1}
A problem has occurred. The system cannot find the account you are trying to access. If you
clicked on the link you received through the verification email, please make sure that your
e-mail client did not break up the link. If so, make sure you correct this problem and retry the
operation.<br>
If the problem persists, please contact <a href="mailto:john.smith@dailynews.com"
>john.smith@dailynews.com</a>.
%]
[%{Status<-1}
[%<%Error%><br>%]
[%A problem occurred while sending the e-mail. The message was: <%SendMailError%><br>%]
The status code for this problem is: <b><%Status%></b><br>
%]
```

Chapter 22 Net Guest

Net Guest enables you to host online interviews. Web guests can send in questions that are answered through the editing interface. The person answering the questions can be anyplace in the world. The person interviewed only needs access to the Editing Component. You should set up a limited account for the guest, which is deactivated after each session.

- NetGuest.pbo
- pbs netguest
  - NetGuestQuestionItem.pbo
- pbs netguest_list
  - NetGuestList.pbo
  - NetGuestListItem.pbo
- pbs netguest_questionform
  - NetGuestQuestionForm.pbo

NetGuest.pbo

<%Caption%>	The caption for the photo.
<%CaptionHeading%>	The heading for the photo.
<%Heading%>	The heading for the interview.
<%Image%>	This field contains 1 if you have an image attached to the interview.
<%ImgLink%>	The URL for the image.
<%ImgWidth%>	The value of the image the width.
<%InfoText%>	The text from the **InfoText** field.
<%Photo%>	The photo credit.
<%Questions%>	This field displays the questions.
<%QuestionEditLink%>	A the link to the Question form.

pbs netguest

<pbs:netguest>

The template is called **gen_netguest.pbs** and is accessed through the address **/apps/pbcs.dll/netguest**.

Options

Count	Controls how many questions are shown. By default, all questions are shown. Enter `count=#` to show a specific number of questions.
Enddate	Sets the end date for the list of questions. By default, all questions are shown. Enter `enddate=YYYYMMDD` to show only questions entered before this date.
Pictureheight	Sets the maximum height of the image.
Picturewidth	Sets the maximum width of the image
Sort	Sets the sort order of the answered questions. By default, the answers are output in the order in which they were answered. Enter `sort=written` to output the answers in the order in which the questions were written.

Sortdirection	Controls the sorting of the questions and answers. By default, the questions and answers are sorted ascending. Enter
	`sortdirection=descending`
	to sort the questions in descending order, with the newest question first. Enter
	`sortdirection=auto`
	if the sort should be descending when the interview is active and ascending when its not.

Start	Sets the start number for the first question. By default the first question is shown. Enter `start=#` and the system will skip the first #-1 questions.

Startdate	Sets the start date for the list of questions. By default, all questions are shown. Enter `startdate=YYYYMMDD` to show only questions entered after this date.

NetGuestQuestionItem.pbo

<%\Author%>	The name of the question submitter.
<%Answer%>	This field contains the answer.
<%Birthday%>	The submitter's birthday.
<%Custom1%>	The information from the first custom field.
<%Custom2%>	The information from the second custom field.
<%\Email%>	The submitter's email.
<%Heading%>	The title for the question.
<%Phone%>	The submitter's phone number.
<%Question%>	The question.

pbs netguest_list

<pbs:netguest_list>

This function is used to list future and past guests. This tag can be placed on all templates. In the example, it is placed on the template **gen_netguest.pbs**.

Options

Archive	Makes the tag list past guests instead of future guests.
Category	Determines the category for which guests will be shown. By default, guests for all categories are shown. To show only guests for the current category enter `category=#category#` This function can also be used to show guests from the current category and subcategory.
Count	Determines how many guests the tag should list. By default, this is 1. To list more guests enter `count=#` where # is a number between 0 and 999. 0 will list all guests. 1 through 999 will list the selected number of guests.
Objectclass	Sets the object class for the object files.
Showref	Controls how many characters of the teaser is shown. By default, the entire teaser is shown.To change the amount of characters from the teaser that is shown enter `showref=#` where # is a number between 1 and 9999.

NetGuestList.pbo

<%Items%>	Outputs the items.

NetGuestListItem.pbo

<%Active%>	This field has the value 1 if the questionlink is active on the guest.
<%\Date%>	Contains the date.
<%Heading%>	Contains the heading.
<%Img%>	This field has the value 1 if there is an image attached to the meeting.
<%ImgLink%>	The link to the image.
<%InfoText%>	The text from the teaser field in the editing interface.
<%IsoDate%>	The date in ISO format.
<%Link%>	The main page.

pbs netguest_questionform

<pbs:netguest_questionform>

The Question form is linked to from the main guest page. The template is called **gen_ngqedit.pbs**.

NetGuestQuestionForm.pbo

This line is used to set up up the form. The action for the form must be

```
<%SaveAction%>
<form action="<%SaveAction%>" method="post">
```

Fields

Birthday	`<input type="text" name="Birthday" size=30>` Captures the birthday.
Custom field 1	`<input type="text" name="Custom1" size=30>` Captures the first custom field. This field can be used for whatever extra information you want to capture.
Custom field 2	`<input type="text" name="Custom2" size=30>` Captures the second custom field. This field can be used for whatever extra information you want to capture.
E-mail address	`<input type="text" name="Email" size=30>` Captures the e-mail address.

Name	`<input type="text" name="Author" size=30>` Captures the name of the user.
Phone	`<input type="text" name="Phone" size=30>` Captures the phone number.
Question	`<textarea name="Question" cols=49 rows=5></textarea>` This field is used for the question.
Title	`<input type="text" name="Heading" size=30>` This field is used for the title.

This hidden field is used by the system to manage the questions:

```
<input type="hidden" name="GID" value="<%GID%>">
```

This field sets the page to which the web guest is sent after submitting the question. You can make this any page. The field `<%NetGuestLink%>` contains the link back to the main Net Guest page:

```
<input type="hidden" name="ReturnPath" value="<%NetGuestLink%>">
```

Chapter 23 User Generated Content

This function allows a web guest to publish a story directly into the SAXOTECH Online database. This feature can be set up to either require registration or login. The stories will appear in the editing interface unpublished on the date the form was submitted. Use the SAXOTECH Online interface to manage the stories.

- pbs userarticleform
  - ArticleForm.pbo
  - ArticleFormMediaItem.pbo
  - ArticleFormSectionItem.pbo
  - UserSubArticles_ArticleFormNotLoggedIn
- pbs usersubmitform for user generated content
  - UserSubmitForm.pbo
  - UserSubmitFormArticleMediaItem.pbo
  - UserSubmitFormArticleSectionItem.pbo
  - UserSubmitFormCategoryList.pbo
    - UserSubmitFormCategoryListItem.pbo
  - UserSubmitFormCCEFieldList.pbo
    - UserSubmitFormCCEFieldListItem.pbo
  - UserSubmitFormCCEModule.pbo
  - UserSubmitFormCCERecurrencePattern.pbo
  - UserSubmitFormErrorFailed.pbo
  - UserSubmitFormErrorInvalidInput.pbo
  - UserSubmitFormErrorNotCreator.pbo
  - UserSubmitFormErrorNotLoggedIn.pbo
  - UserSubmitFormExtraFields.pbo
    - UserSubmitFormExtraFieldsItem.pbo
  - UserSubmitFormPhotoGalleryPhotoItem.pbo
  - UserSubmitFormTaxonomyList.pbo
    - UserSubmitFormTaxonomyListItem.pbo
- UserSubmitFormArticle.pbo
- UserSubmitFormArticleCategoryList.pbo
- UserSubmitFormArticleCategoryListItem.pbo

- UserSubmitFormArticleErrorFailed.pbo
- UserSubmitFormArticleErrorInvalidInput.pbo
- UserSubmitFormArticleErrorNotCreator.pbo
- UserSubmitFormArticleErrorNotLoggedIn.pbo
- UserSubmitFormArticleExtraFields.pbo
- UserSubmitFormArticleExtraFieldsItem.pbo
- UserSubmitFormArticleTaxonomyList.pbo
- UserSubmitFormArticleTaxonomyListItem.pbo
- UserSubmitFormCCE.pbo
- UserSubmitFormCCECategoryList.pbo
- UserSubmitFormCCECategoryListItem.pbo
- UserSubmitFormCCEErrorFailed.pbo
- UserSubmitFormCCEErrorInvalidInput.pbo
- UserSubmitFormCCEErrorNotCreator.pbo
- UserSubmitFormCCEErrorNotLoggedIn.pbo
- UserSubmitFormCCERecurrencePattern.pbon
- UserSubmitFormCCETaxonomyList.pbo
- UserSubmitFormCCETaxonomyListItem.pbom
- UserSubmitFormPhotoGallery.pbo
- UserSubmitFormPhotoGalleryCategoryList.pbo
- UserSubmitFormPhotoGalleryCategoryListItem.pbo
- UserSubmitFormPhotoGalleryErrorFailed.pbo
- UserSubmitFormPhotoGalleryErrorInvalidInput.pbo
- UserSubmitFormPhotoGalleryErrorNotCreator.pbo
- UserSubmitFormPhotoGalleryErrorNotLoggedIn.pbo
- UserSubmitFormPhotoGalleryExtraFields.pbo
- UserSubmitFormPhotoGalleryExtraFieldsItem.pbo
- UserSubmitFormPhotoGalleryTaxonomyList.pbo
- UserSubmitFormPhotoGalleryTaxonomyListItem.pbo
- User Submitted Stories

From version 6.2.0 the feature has been completely reworked. The existing object files have been renamed, giving them a more general prefix because it is now possible to submit items other than stories. The system will fallback to the pre-6.2.0 object filenames for backward compatibility. However, they should be considered deprecated.

\

- ArticleForm.pbo Renamed to UserSubmitFormArticle.pbo
- ArticleFormMediaItem.pbo Renamed to UserSubmitFormArticleMediaItem.pbo
- ArticleFormNotLoggedIn.pbo Renamed to UserSubmitFormArticleErrorNotLoggedIn.pbo
- ArticleFormSectionItem.pbo Renamed to UserSubmitFormArticleSectionItem.pbo

- <pbs:userarticleform> Renamed to <pbs:UserSubmitForm>

The new section

The tag <Pbs:UserSubmitForm> generates a submit form formatted through a number of object files and can load in an existing item for updating purposes.

The object files are listed below and they allow specialization through the filename. For example, generating a story submit form will first try to retrieve the file prefix *UserSubmitFormArticle*. If it is not available, it will try the file prefix *UserSubmitForm*. Finally, if applicable it will try the depricated file above.

The following object files can display stories, photo galleries, and CCE objects:

UserSubmitForm.pbo
UserSubmitFormTaxonomyList.pbo
UserSubmitFormTaxonomyListItem.pbo

UserSubmitFormErrorNotLoggedIn.pbo
UserSubmitFormErrorNotCreator.pbo
UserSubmitFormErrorInvalidInput.pbo
UserSubmitFormErrorFailed.pbo

The following general object files are only applicable for displaying stories and photo galleries:

UserSubmitFormCategoryList.pbo
UserSubmitFormCategoryListItem.pbo
UserSubmitFormExtraFields.pbo
UserSubmitFormExtraFieldsItem.pbo
Live-Extrafields\<FieldName>.pbo

The following object files are available for stories:

UserSubmitFormArticle.pbo
UserSubmitFormArticleCategoryList.pbo

UserSubmitFormArticleCategoryListItem.pbo
UserSubmitFormArticleTaxonomyList.pbo
UserSubmitFormArticleTaxonomyListItem.pbo
UserSubmitFormArticleExtraFields.pbo
UserSubmitFormArticleExtraFieldsItem.pbo
UserSubmitFormArticleMediaItem.pbo
UserSubmitFormArticleSectionItem.pbo

UserSubmitFormArticleErrorNotLoggedIn.pbo
UserSubmitFormArticleErrorNotCreator.pbo
UserSubmitFormArticleErrorInvalidInput.pbo
UserSubmitFormArticleErrorFailed.pbo

The following object files are available for photo galleries:

UserSubmitFormPhotoGallery.pbo
UserSubmitFormPhotoGalleryCategoryList.pbo
UserSubmitFormPhotoGalleryCategoryListItem.pbo
UserSubmitFormPhotoGalleryTaxonomyList.pbo
UserSubmitFormPhotoGalleryTaxonomyListItem.pbo
UserSubmitFormPhotoGalleryExtraFields.pbo
UserSubmitFormPhotoGalleryExtraFieldsItem.pbo
UserSubmitFormPhotoGalleryPhotoItem.pbo

UserSubmitFormPhotoGalleryErrorNotLoggedIn.pbo
UserSubmitFormPhotoGalleryErrorNotCreator.pbo
UserSubmitFormPhotoGalleryErrorInvalidInput.pbo
UserSubmitFormPhotoGalleryErrorFailed.pbo

The following object files are available for CCE objects:

UserSubmitFormCCE.pbo
UserSubmitFormCCECategoryList.pbo
UserSubmitFormCCECategoryListItem.pbo
UserSubmitFormCCETaxonomyList.pbo
UserSubmitFormCCETaxonomyListItem.pbo
UserSubmitFormCCEModule.pbo
UserSubmitFormCCEFieldList.pbo
UserSubmitFormCCEFieldListItem.pbo
UserSubmitFormCCERecurrencePattern.pbo

UserSubmitFormCCEErrorNotLoggedIn.pbo
UserSubmitFormCCEErrorNotCreator.pbo
UserSubmitFormCCEErrorInvalidInput.pbo
UserSubmitFormCCEErrorFailed.pbo

pbs userarticleform

<pbs:userarticleform>

This tag is used to display the form.

A form is required for users to submit stories. The form requires that the user is logged in. You can either set the security option on the category by setting the subscription type on the category, or you can set the category to require registration by adding `&Login=1` to the link.

Template

This function should be used on a section template. The example is placed on Usersubmit_Ovr.pbs.

Options

Mediafiles	Allows multiple asset files to be submitted with a story. Enter `mediafiles=x` where x indicates the number of possible assets that can be uploaded.
Objectclass	Sets the object class used.

Sections	Controls how many paragraphs the page will contain. By default, 7 paragraph boxes are displayed. Enter
	`sections=x`
	to have
	`x`
	number of paragraph boxes on the page.

Useobjects	Enables the use of object files.

ArticleForm.pbo

The object file contains the following fields:

Byline	Writes the byline to the database. It is used like this: `<input type="text" size="50" name="byline">`

Category	Determines the category in which the story will be saved. It is used like this:
	`<select name="category">`
	`<option value="LETTERS">Letters`
	`<option value="OPINION">Opinion`
	`</select>`

<%Counter%>	The counter for the paragraphs.
<%\Date%>	The current date. It can be used in conjunction with the date input field like this: `<b>Date:</b> <input type="text" size="10" name="date" value="<%\Date%>">`
Location	Writes the location to the database. It is used like this: `<input type="text" size="30" name="location"> `
<%mediafiles%>	Outputs the media files, such as images.
Picture	The local path to a picture that will be uploaded. It is used like this: `<input type="file" size="30" name="picture" value=" Picture... ">`
Picturebyline	The picture byline. `<input type="text" size="50" name="picturebyline">`

Picturecaption	The picture caption. `<textarea cols="50" rows="3" name="picturecaption"></textarea> `
Picturecaptionheading	The picture caption heading. `<input type="text" size="50" name="picturecaptionheading">`

<%PostAction%>	Contains the URL used by the submission form. It is used like this:
	`<form action="<%PostAction%>" method="POST" enctype="multipart/form-data"`
	`>`
	`<input type="hidden" name="status" value="1">`

RedirectURL	Sets the redirection page to which the web guest is sent after submitting the form. The field is used like this:
	`<input type="hidden" name="RedirectURL" value=`
	`"/apps/pbcs.dll/section?category=SUB">`

<%Sections%>	Outputs the paragraphs.
Status	This field can be used to override the story status. By default, stories come in as unpublished. If you would like to publish automatically upon submission, enter the following line in the form: `<input type="hidden" name="status" value="9">`
Subtitle	Writes the subtitle to the database. It is used like this: `<input type="text" size="66" name="title2">`
Summary	Writes the summary to the database. It is used like this: `<textarea cols="50" rows="7" name="summary"></textarea>`
Superheading	Writes the Superheading to the database. It is used like this: `<input type="text" size="66" name="superheading">`
Title	Writes the title to the database. It is used like this: `<input type="text" size="66" name="title">`
<%UserEmail%>	The logged in user's email address.
<%UserName%>	The logged in user's first and last name.

ArticleFormMediaItem.pbo

The ArticleFormMediaItem.pbo is used when rendering the submit form. It must contain the following:

name="media<%Counter%>"	Selects the media file.
name="media<%Counter%>description"	Used for a description of the media.

The object file has the following pbo variables:

<%counter%>	Returns the number of assets.
<%NoOfSections%>	Returns the value of `<pbs:userarticleform>` tag option "mediafiles."

ArticleFormSectionItem.pbo

The object file contains the following fields:

body<%Counter%>	The paragraph body. It is used like this: `<textarea cols="50" rows="7" name="body<%Counter%>" ></textarea>`
body<%Counter%>image	This selects the image file to be used with the paragraph. It is used like this:
body<%Counter%>imagebyline	The byline to be used with the selected image in the paragraph. It is used like this:
body<%Counter%>imagecaption	The caption to be used with the selected image in the paragraph. It is used like this:
body<%Counter%>imagetext	Any text to be used with the selected image in the paragraph. It is used like this:
intermediateheading<%Counter%>	The paragraph heading. It is used like this: `<input type="text" size="66" name= "intermediateheading<%Counter%>">`

This ArticleFormSectionItem.pbo file has the following pbo variables:

<%counter%>	Returns the paragraph number
<%NoOfSections%>	This variable returns the value of the `<pbs:userarticleform>` tag option "sections."

UserSubArticles_ArticleFormNotLoggedIn

ArticleFormNotLoggedIn.pbo

This object file contains no fields.

pbs usersubmitform for user generated content

<pbs:usersubmitform>

From version 6.2.0

The tag is used to submit data to the system through the front-end. The data can be a story, a photo gallery or a CCE object, and the data type is determined through the parameter of **SubmitType**.

Parameters available

ObjectClass	Specifies what object file design class to use.
SubmitType	Specifies what type of data is to be submitted. Use 'ph' for photo gallery, 'cce' for CCE object, and 'ar' for story. Default is 'ar'.

Stories

Every story specific form tag supports the following parameters

Sections	Defines the maximum number of paragraphs to be available to the user.
MediaFiles	Defines the maximum number of media files that can be uploaded by the user.

Photo galleries

Every gallery specific form tag supports the following parameters

Sections	Same as **MaxAllowedPhotos**.
MaxAllowedPhotos	Defines the maximum number of images to be available to the user.

CCE objects

Every CCE object specific form tag has to specify a specific Module, Class, and Type that the CCE object will belong to.

CCESection	Specifies Module, Class, and Type that the CCE object will belong to on the form "<ModuleID><*ClassID*><TypeID>". Matching settings in the publicus.ini file are required for the user to be able to do a succesful submit.

Example:

```
<pbs:usersubmitform ObjectClass="1" CCESection="2_12_Cars" SubmitType="CCE">
```

Updating an existing item

If an existing item is to be loaded into the form so that the item is updated on save, the URL parameters to the call that fetches the `<pbs:UserSubmitForm>` tag contents must contain **load=1** and the **ID of the item to be loaded**.

For stories the following ID parameters are required:

```
/apps/pbcs.dll/section?category=storysubmit&OrgDate=X&OrgCategory=Y&OrgStoryID=Z&load=1
```

Where X, Y, and Z values match a story within the system.

For photo galleries the following ID parameters are required:

```
/apps/pbcs.dll/section?category=gallerysubmit&OrgDate=X&OrgCategory=Y&OrgStoryID=Z&load=1
```

Where X, Y, and Z values match a photo gallery within the system.

For CCE objects the following ID parameter is required:

```
/apps/pbcs.dll/section?category=ccesubmit&OrgCCEID=X&load=1
```

Where the value of X matches a CCE object within the system.

UserSubmitForm.pbo

From version 6.2.0

Object file for formatting the submit form for the user generated by <pbs:UserSubmitForm>.The object files of UserSubmitFormArticle.pbo , UserSubmitFormPhotoGallery.pbo, and UserSubmitFormCCE.pbo are also available for the specific **SubmitType**.

The following general variables are available for all submit types:

<%Date%>	Contains the current date on the form YYYYMMDD.
<%PostAction%>	Contains the URL the form is to be submitted to.
<%SubmitType%>	Contains what the form is to submit.
<%TaxonomyWords%>	Contains the taxonomy words available to the user formatting using the object file of UserSubmitFormTaxonomyList.pbo
<%UserID%>	The user's ID. If the user is anonymous the value matches the AnonymousUserId setting.
<%UserName%>	The user's full name, e.g., John Doe. If the user is anonymous the variable is empty.
<%UserEmail%>	The user's email address. If the user is anonymous the variable is empty.

Also, there are Story specific, Photo gallery specific, and CCE specific variables.

Story specific variables

<%Categories%>	Contains the available categories formatted using the object file of UserSubmitFormCategoryList.pbo
<%ExtraFields%>	Contains the extra fields formatted using the object file of UserSubmitFormExtraFields.pbo
<%ExtraFieldType%>	Contains the name of the extra fields' group name.
<%MediaFiles%>	Contains the media files input fields formatted using the object file of UserSubmitFormArticleMediaItem.pbo
<%NoOfMediaFiles%>	Contains the number of allowed media files in the form based on the tag parameter of **MediaFiles**.
<%Sections%>	Contains the story paragraphs formatted using UserSubmitFormArticleSectionItem.pbo
<%NoOfSections%>	Contains the number of paragraphs.

When updating an existing story, the following variables are available as well:

<%Status%>	Contains the published status of the story. 1 is unpublished, 9 is published.
<%ArticleStatus%>	Same as **<%Status%>**
<%Summary%>	Contains the summary of the story.
<%Title%>	Contains the title of the story.
<%Location%>	Contains the location of the story.

<%PictureCaption%>	Contains the caption of the main image.
<%PictureSummary%>	Contains the image summary of the main image.
<%PictureByline%>	Contains the image byline of the main image.
<%PictureURL%>	Contains a link to the main image.

<%OrgSections%>	Contains the number of paragraphs in the original.
<%OrgDate%>	Contains the original date.
<%OrgStoryID%>	Contains the original ID.
<%OrgCategory%>	Contains the original category.

Photo gallery specific variables

<%Categories%>	Contains the available categories formatted using the object file of UserSubmitFormCategoryList.pbo
<%ExtraFields%>	Contains the extra fields formatted using the object file of UserSubmitFormExtraFields.pbo
<%ExtraFieldType%>	Contains the name of the extra fields' group name.
<%Sections%>	Contains the gallery images formatted using UserSubmitFormPhotoGalleryPhotoItem.pbo
<%NoOfSections%>	Contains the number of images.

When updating an existing gallery, the following variables are available as well:

<%Status%>	Contains the published status of the gallery. 0 is unpublished, 1 is published.
<%PhotoGalleryStatus%>	Same as **<%Status%>**
<%Summary%>	Contains the summary of the gallery.
<%Title%>	Contains the title of the gallery.
<%Location%>	Contains the location of the gallery.

<%OrgSections%>	Contains the number of images in the original.
<%OrgDate%>	Contains the original date.
<%OrgStoryID%>	Contains the original ID.
<%OrgCategory%>	Contains the original category.

CCE object specific variables

<%Module%>	Contains the module and type information formatted using the object file of UserSubmitFormCCEModule.pbo
<%ShowRecurrencePattern%>	Contains whether the object to be submitted belongs to an Event module (=1) or not (=0)
<%Fields%>	Contains the extra fields formatted using the object file of **UserSubmitFormCCEFieldList.pbo**

If **<%ShowRecurrencePattern%>** is 1 then the following variables are available as well:

<%CCERecurrencePattern%>	Contains the recurrence pattern formatted using the object file of UserSubmitFormCCERecurrencePattern.pbo
<%Pattern%>	Contains the overall recurrence pattern used for calculating when the object event occurs.
<%StopMethod%>	Contains the method used for determining when the event series stops. 0 means a set date, 1 means after a certain amount of occurrences.

If editing an existing object, the following variables are available:

<%Latitude%>	Contains the latitude of the object.
<%Longitude%>	Contains the longitude of the object.
<%PictureURL%>	Contains the image link to the object's attached image if there is one.
<%RowGUID%>	The ID of the object.
<%Summary%>	Contains the summary of the object.
<%Title%>	Contains the title of the object.

UserSubmitFormArticleMediaItem.pbo

From version 6.2.0

Object file for formatting the media files of the story submit form for the user generated by **<pbs:UserSubmitForm SubmitType="ar">**.

<%Count%>	A forthrunning number.
<%NoOfMediaFiles%>	Contains the number of media files allowed by the **MediaFiles** tag parameter.

UserSubmitFormArticleSectionItem.pbo

From version 6.2.0

Object file for formatting the story paragraphs of the submit form for the user generated by `<pbs:UserSubmitForm SubmitType="ar">`.

<%Counter%>	A forthrunning number.
<%NoOfSections%>	Contains the maximum number of paragraphs available, determined by the **Sections** tag parameter.
<%OrgSections%>	Contains the number of paragraphs on the original story when updating it - otherwise 0.
<%HasContent%>	Determines whether the paragraphs contains an image (=1) or not (=0).

In addition, if updating a story the following variables are available:

<%Body%>	Contains the paragraph text.
<%PictureCaption%>	Contains the paragraph image title.
<%PictureByline%>	Contains the paragraph image byline.
<%PictureSummary%>	Contains the paragraph image text.
<%PictureUrl%>	Contains the URL to the paragraph image.

UserSubmitFormCategoryList.pbo

From version 6.2.0

Object file for formatting the category list part of the submit form for the user generated by <pbs:UserSubmitForm>.The object files UserSubmitFormArticleCategoryList.pbo and UserSubmitFormPhotoGalleryCategoryList.pbo are also available for the specific **SubmitType**.

<%Items%>	Contains the available categories formatted using UserSubmitFormCategoryListItem.pbo
<%Count%>	Contains the amount of categories available.

UserSubmitFormCategoryListItem.pbo

From version 6.2.0

Object file for formatting the category part of the submit form for the user generated by <pbs:UserSubmitForm>.The object files UserSubmitFormArticleCategoryListItem.pbo and UserSubmitFormPhotoGalleryCategoryListItem.pbo are also available for the specific **SubmitType**.

<%ID%>	Contains the category ID.
<%Count%>	Forthrunning number.
<%Name%>	Contains the display text.
<%Selected%>	Has the value 'selected' if the item has the category as category. Only available when updating existing data.

UserSubmitFormCCEFieldList.pbo

From version 6.2.0

Object file for formatting the CCE fields part of the submit form for the user generated by **<pbs:UserSubmitForm SubmitType="cce">**.

The following variables are available:

<%Items%>	Contains the fields, formatted using the object file UserSubmitFormCCEFieldListItem.pbo
<%NoOfFields%>	Contains the number of fields in **<%Items%>**.

UserSubmitFormCCEFieldListItem.pbo

From version 6.2.0

Object file for formatting the CCE fields of the submit form for the user generated by **<pbs:UserSubmitForm SubmitType="cce">**.

The following variables are available:

<%Counter%>	Contains the the field number, starting from 1.
<%DataType%>	Contains the data type of the field.
<%DisplayName%>	Contains the display name.
<%Name%>	Contains the internal name.
<%Value%>	Contains the value of the field when updating an existing object.

UserSubmitFormCCEModule.pbo

From version 6.2.0

Object file for formatting the module and type information part of the submit form for the user generated by **<pbs:UserSubmitForm SubmitType="cce">**.

The following variables are available:

<%ModuleID%>	The module ID.
<%ModuleName%>	The module name.
<%ModuleShortName%>	The module short name.

If the CCE object belongs to a class, the following variables are available:

<%ClassID%>	The class ID.
<%ClassName%>	The class name.

If the CCE object belongs to a type, the following variables are available:

<%TypeID%>	The type ID.
<%TypeName%>	The type name.

UserSubmitFormCCERecurrencePattern.pbo

From version 6.2.0

Object file for formatting the recurrence pattern part of the submit form for the user generated by **<pbs:UserSubmitForm**

SubmitType="cce">. It is only used for objects belonging to Event modules.

The following variables are available when updating a CCE event:

<%FromDate%>	Contains from what date the system is to calculate occurrences (yyyymmdd format)
<%ToDate%>	Contains to what date the system is to calculate occurrences (yyyymmdd format) - it is used when **StopMethod** is 0.
<%NumberOfOcc%>	Contains how many occurrences to calculate from the **FromDate** - it is used when **StopMethod** is 1.
<%StopMethod%>	Contains how the system is to stop calculating occurrences. 0 is on a set date, 1 is after a certain number of occurrences.
<%FromTime%>	Contains from what time of day the object occurs.
<%ToTime%>	Contains to what time of day the object occurs.
<%Pattern%>	Contains the overall method used to calculate occurrences.
<%Subpattern%>	Contains the specific method used to calculate occurrences.
<%Jump1%>	Contains the first jump value in the pattern. E.g. 1 in "The first Monday in June".
<%Jump2%>	Contains the second jump value in the pattern. E.g., 6 in "The first Monday in June".
<%Day1%>	If Sunday is selected in the pattern then it has value 1, otherwise it has no value.
<%Day2%>	If Monday is selected in the pattern then it has value 1, otherwise it has no value.
<%Day3%>	If Tuesday is selected in the pattern then it has value 1, otherwise it has no value.
<%Day4%>	If Wednesday´ is selected in the pattern then it has value 1, otherwise it has no value.
<%Day5%>	If Thurday is selected in the pattern then it has value 1, otherwise it has no value.
<%Day6%>	If Friday is selected in the pattern then it has value 1, otherwise it has no value.
<%Day7%>	If Saturday is selected in the pattern then it has value 1, otherwise it has no value.

Recurrence pattern description

The pattern used to calculate object occurrences is made up by a main pattern and then a subpattern, for our purposes lets denote the full pattern by the **<%Pattern%>**,**<%SubPattern%>** values. Then the following patterns are available:

1 - Daily pattern

1,1 = For every <Jump1> days
1,2 = Every weekday (from Monday to Friday)

2 - Weekly pattern

2,1 = Repeat for every <Jump1> weeks on <DayX,Y,Z>

where <DayX,Y,Z> is actually specified by having the value set to 1 for any number of the following fields:
<Day1>,<Day2>,<Day3>,<Day4>,<Day5>,<Day6>,<Day7>

3 - Monthly pattern

3,1 = On day <Jump1> every <Jump2> months
3,2 = The First..Fourth,Last(=5) (<Jump1>) <DayN> every <Jump2> months

where <DayN> is actually specified by having the value set to 1 for one of the following fields:
<Day1>,<Day2>,<Day3>,<Day4>,<Day5>,<Day6>,<Day7>

If more than one is selected, its the first selected day in the week that is the actual one being selected.

4 - Yearly pattern

4,1 = On day <Jump1> in Month <Jump2>
4,2 = The First..Fourth,Last(=5) (<Jump1>) <DayN> in Month <Jump2>

where <DayN> is actually specified by having the value set to 1 for one of the following fields:
<Day1>,<Day2>,<Day3>,<Day4>,<Day5>,<Day6>,<Day7>

If more than one is selected, its the first selected day in the week that is the actual one being selected.

Examples:

FromDate may not be the first day in the pattern, e.g., when selecting only Mondays and today is a Tuesday, the first event date is actually 6 days in the future. The following examples show how patterns selects occurrence days

1 - Daily

1,1 = For every 2 days - If **FromDate** is a Monday, then it selects Monday, Wednesday, Friday etc.

1,2 = Selection is defined as weekdays - if **FromDate** is a Thursday, it selects Thursday, Friday, Monday, Tuesday, Wednesday, Thursday etc.

2 - Weekly

2,1 = Repeat for every 2 weeks on Mondays and Tuesdays - if **FromDate** is a Tuesday, that day is selected. The next weeks' Monday and Tuesday is not selected. In the week after that week the Monday and Tuesday is selected. Then two weeks later Monday and Tuesday is selected again.

2,1 = Repeat for every 2 weeks on Mondays and Tuesdays - if **FromDate** is a Wednesday, no selections are made for the week. The next weeks' Monday and Tuesday is not selected. In the week after that week the Monday and Tuesday is selected. Then two weeks later Monday and Tuesday is selected again.

3 - Monthly

3,1 = On day 14 every 2 months - if **FromDate** is February 13th, it selects February 14th, April 14th, June 14th etc.

3,1 = On day 14 every 2 months - if **FromDate** is February 17th, it selects April 14th, June 14th etc.

3,2 = The second Tuesday every 1 month. If **FromDate** is February 17th, 2009, the second Tuesday is the 10th. Therefore, the first occurrence is the second Tuesday of March, second occurrence is the second Tuesday in April etc. Had the second Tuesday in the current month been in the future, then that Tuesday would have been the first event date.

4 - Yearly

4,1 = On day 14 in month 2 - if **FromDate** is February 17th, 2009 the first occurrence is February 14th, 2010. Had the **FromDate** been a week before, the first occurrence had been on February 14th, 2009.

4,2 = The last Sunday in month 2 - if **FromDate** is before the given date, the first occurrence is in the same year, otherwise they start being calculated from the next.

Form examples

Submit a simple non-recurring events

Here we want a simple input form, where the event enterer only needs to supply date and time of the event.

Because we don't want any recurrence to happen, we preselect the recurrence pattern to stop after one occurrence. And the pattern we choose is the daily pattern (1,1) with jump 1 (every day) so that we know for sure that the actual event date is going to be the same day as the start date.

```
  <input type="hidden" name="stopmethod" value="1" />
  <input type="hidden" name="numberofocc" value="1" />
  <input type="hidden" name="pattern" value="1" />
  <input type="hidden" name="pattern1sub" value="1" />
  <input type="hidden" name="pattern1sub1jump1" value="1" />

  <tr>
    <td>
      <table cellpadding=1 cellspacing=0 border=0>

        <tr>
          <td>
            Date of the event:
          </td>
          <td>
            <input name="fromdate" id="fromdate" maxlength=8 value="<%FromDate%>" /> (yyyymmdd)
          </td>
        </tr>

        <tr>
          <td>
            Start time:
          </td>
          <td>
            <input name="fromtime" id="fromtime" maxlength=4 value="<%FromTime%>" /> (HHMM,
24h)
          </td>
        </tr>

        <tr>
          <td>
            End time:
          </td>
          <td>
            <input name="totime" id="totime" maxlength=4 value="<%ToTime%>" /> (HHMM, 24h)
          </td>
        </tr>

      </table>
    </td>
  </tr>
```

UserSubmitFormErrorFailed.pbo

From version 6.2.0

Object file for formatting the response where something unexpected happened.

The object files UserSubmitFormArticleErrorFailed.pbo, UserSubmitFormPhotoGalleryErrorFailed.pbo, and UserSubmitFormCCEErrorFailed.pbo are also available for the specific **SubmitType**.

<%ErrorMessage%>	Contains an error message

UserSubmitFormErrorInvalidInput.pbo

From version 6.2.0

Object file for formatting the response for the user trying to submit a form but with invalid data.

The object files of UserSubmitFormArticleErrorInvalidInput.pbo, UserSubmitFormPhotoGalleryErrorInvalidInput.pbo, and UserSubmitFormCCEErrorInvalidInput.pbo are also available for the specific **SubmitType**.

<%ErrorMessage%>	Contains an error message
<%ErrorField%>	Contains the name of the field containing invalid data.

UserSubmitFormErrorNotCreator.pbo

From version 6.2.0

Object file for formatting the response for the user trying to submit a form holding an existing item that is being updated and where the user is is not the creator of the item.

The object files UserSubmitFormArticleErrorNotCreator.pbo, UserSubmitFormPhotoGalleryErrorNotCreator.pbo, and UserSubmitFormCCEErrorNotCreator.pbo are also available for the specific **SubmitType**.

<%ErrorMessage%>	Contains an error message.
<%ErrorField%>	Contains the value 'User'.

UserSubmitFormErrorNotLoggedIn.pbo

From version 6.2.0

Object file for formatting the response for the user trying to submit a form but where the user is not logged in.

The object files of UserSubmitFormArticleErrorNotLoggedIn.pbo, UserSubmitFormPhotoGalleryErrorNotLoggedIn.pbo, and UserSubmitFormCCEErrorNotLoggedIn.pbo are also available for the specific **SubmitType**.

<%ErrorMessage%>	Contains an error message.
<%ErrorField%>	Contains the value 'User'.

UserSubmitFormExtraFields.pbo

From version 6.2.0

Object file for formatting the extra fields list part of the submit form for the user generated by pbs usersubmitform. The object files of UserSubmitFormArticleExtraFields.pbo and UserSubmitFormPhotoGalleryExtraFields.pbo are also available for the specific **SubmitType** .

<%Items%>	Contains the extra fields formatted using **UserSubmitFormExtraFields.pbo**.
<%Count%>	Contains the amount of extra fields.

UserSubmitFormExtraFieldsItem.pbo

From version 6.2.0

Object file for formatting the extra fields part of the submit form for the user generated by <pbs:UserSubmitForm>. The object files UserSubmitFormArticleExtraFieldsItem.pbo and UserSubmitFormPhotoGalleryExtraFieldsItem.pbo are also available for the specific **SubmitType**.

In addition, field specific object files are available at **/Objects/Live-Extrafields/<FieldName>.pbo**

<%ID%>	Contains the extra field ID.
<%Count%>	A forthrunning number.
<%Name%>	Contains the extra field display text.
<%Value%>	Has the value of the extra field. Only available when updating existing data.

UserSubmitFormPhotoGalleryPhotoItem.pbo

From version 6.2.0

Object file for formatting the gallery image of the submit form for the user generated by **<pbs:UserSubmitForm SubmitType="ph">**.

<%Counter%>	A forthrunning number.
<%NoOfSections%>	Contains the maximum number of paragraphs available, determined by the **Sections** tag parameter.
<%OrgSections%>	Contains the number of paragraphs on the original story when updating it - otherwise 0.
<%HasContent%>	Contains whether the paragraphs contains an image (=1) or not (=0).

In addition, if updating a gallery the following variables are available:

<%PictureCaption%>	Contains the image title.
<%PictureByline%>	Contains the image byline.
<%PictureSummary%>	Contains the image text.
<%PictureUrl%>	Contains the URL to the image.

UserSubmitFormTaxonomyList.pbo

From version 6.2.0

Object file for formatting the taxonomy words list part of the submit form for the user generated by <pbs:UserSubmitForm>.The object files UserSubmitFormArticleTaxonomyList.pbo, UserSubmitFormPhotoGalleryTaxonomyList.pbo, and UserSubmitFormCCETaxonomyList.pbo are also available for the specific **SubmitType**.

<%Items%>	Contains the taxonomy words formatted using **UserSubmitFormTaxonomyList.pbo**.
<%Count%>	Contains the amount of taxonomy words.

UserSubmitFormTaxonomyListItem.pbo

From version 6.2.0

Object file for formatting the taxonomy words part of the submit form for the user generated by <pbs:UserSubmitForm>.The object files of UserSubmitFormArticleTaxonomyListItem.pbo, UserSubmitFormPhotoGalleryTaxonomyListItem.pbo, and UserSubmitFormCCETaxonomyListItem.pbo are also available for the specific **SubmitType**.

<%ID%>	Contains the taxonomy word ID.
<%Count%>	A forthrunning number.
<%Name%>	Contains the taxonomy word display text.
<%Selected%>	Has the value 'selected' if the item is associated with the taxonomy word. Only available when updating existing data.

UserSubmitFormArticle.pbo

Works the same way as UserSubmitForm.pbo.

UserSubmitFormArticleCategoryList.pbo

Works the same way as UserSubmitFormCategoryList.pbo.

UserSubmitFormArticleCategoryListItem.pbo

Works the same way as UserSubmitFormCategoryListItem.pbo.

UserSubmitFormArticleErrorFailed.pbo

Works the same way as UserSubmitFormErrorFailed.pbo.

UserSubmitFormArticleErrorInvalidInput.pbo

Works the same way as UserSubmitFormErrorInvalidInput.pbo.

UserSubmitFormArticleErrorNotCreator.pbo

Works the same way as UserSubmitFormErrorNotCreator.pbo.

UserSubmitFormArticleErrorNotLoggedIn.pbo

Works the same way as UserSubmitFormErrorNotLoggedIn.pbo.

UserSubmitFormArticleExtraFields.pbo

Works the same way as UserSubmitFormExtraFields.pbo.

UserSubmitFormArticleExtraFieldsItem.pbo

Works the same way as UserSubmitFormExtraFieldsItem.pbo.

UserSubmitFormArticleTaxonomyList.pbo

Works the same way as UserSubmitFormTaxonomyList.pbo.

UserSubmitFormArticleTaxonomyListItem.pbo

Works the same way as UserSubmitFormTaxonomyListItem.pbo.

UserSubmitFormCCE.pbo

Works the same way as UserSubmitForm.pbo.

UserSubmitFormCCECategoryList.pbo

Works the same way as UserSubmitFormCategoryList.pbo.

UserSubmitFormCCECategoryListItem.pbo

Works the same way as UserSubmitFormCategoryListItem.pbo.

UserSubmitFormCCEErrorFailed.pbo

Works the same way as UserSubmitFormErrorFailed.pbo.

UserSubmitFormCCEErrorInvalidInput.pbo

Works the same way as UserSubmitFormErrorInvalidInput.pbo.

UserSubmitFormCCEErrorNotCreator.pbo

Works the same way as UserSubmitFormErrorNotCreator.pbo.

UserSubmitFormCCEErrorNotLoggedIn.pbo

Works the same way as UserSubmitFormErrorNotLoggedIn.pbo.

UserSubmitFormCCERecurrencePattern.pbon

Works the same way as UserSubmitFormRecurrencePattern.pbo.

UserSubmitFormCCETaxonomyList.pbo

Works the same way as UserSubmitFormTaxonomyList.pbo.

UserSubmitFormCCETaxonomyListItem.pbom

Works the same way as UserSubmitFormTaxonomyListItem.pbo.

UserSubmitFormPhotoGallery.pbo

Works the same way as UserSubmitForm.pbo.

UserSubmitFormPhotoGalleryCategoryList.pbo

Works the same way as UserSubmitFormCategoryList.pbo.

UserSubmitFormPhotoGalleryCategoryListItem.pbo

Works the same way as UserSubmitFormCategoryListItem.pbo.

UserSubmitFormPhotoGalleryErrorFailed.pbo

Works the same way as UserSubmitFormErrorFailed.pbo.

UserSubmitFormPhotoGalleryErrorInvalidInput.pbo

Works the same way as UserSubmitFormErrorInvalidInput.pbo.

UserSubmitFormPhotoGalleryErrorNotCreator.pbo

Works the same way as UserSubmitFormErrorNotCreator.pbo.

UserSubmitFormPhotoGalleryErrorNotLoggedIn.pbo

Works the same way as UserSubmitFormErrorNotLoggedIn.pbo.

UserSubmitFormPhotoGalleryExtraFields.pbo

Works the same way as UserSubmitFormExtraFields.pbo.

UserSubmitFormPhotoGalleryExtraFieldsItem.pbo

Works the same way as UserSubmitFormExtraFieldsItem.pbo.

UserSubmitFormPhotoGalleryTaxonomyList.pbo

Works the same way as UserSubmitFormTaxonomyList.pbo.

UserSubmitFormPhotoGalleryTaxonomyListItem.pbo

Works the same way as UserSubmitFormTaxonomyListItem.pbo.

User Submitted Stories

This function allows a web guest to publish a story directly into the SAXOTECH Online database. This feature can be set up to either require registration or login. The stories will appear in the editing interface unpublished on the date the form was submitted. Use the SAXOTECH Online interface to manage the stories.

The following object files can display stories:

UserSubmitForm.pbo
UserSubmitFormTaxonomyList.pbo
UserSubmitFormTaxonomyListItem.pbo
UserSubmitFormErrorNotLoggedIn.pbo
UserSubmitFormErrorNotCreator.pbo
UserSubmitFormErrorInvalidInput.pbo
UserSubmitFormErrorFailed.pbo

The following general object files are also available for displaying stories:

UserSubmitFormCategoryList.pbo
UserSubmitFormCategoryListItem.pbo
UserSubmitFormExtraFields.pbo
UserSubmitFormExtraFieldsItem.pbo
Live-Extrafields\<FieldName>.pbo

The following object files are available for stories:

UserSubmitFormArticle.pbo
UserSubmitFormArticleCategoryList.pbo
UserSubmitFormArticleCategoryListItem.pbo
UserSubmitFormArticleTaxonomyList.pbo
UserSubmitFormArticleTaxonomyListItem.pbo
UserSubmitFormArticleExtraFields.pbo
UserSubmitFormArticleExtraFieldsItem.pbo
UserSubmitFormArticleMediaItem.pbo
UserSubmitFormArticleSectionItem.pbo

UserSubmitFormArticleErrorNotLoggedIn.pbo
UserSubmitFormArticleErrorNotCreator.pbo
UserSubmitFormArticleErrorInvalidInput.pbo
UserSubmitFormArticleErrorFailed.pbo

Chapter 24 Subscriptions

Subscriptions can be set on all categories. When subscriptions are enabled, you need to have the template gen_login.pbs to handle logins. Subscriptions are a separate function from registration.

User information is stored in the SAXOTECH Online database. The user management screen is documented in the Subscriptions chapter in the SAXOTECH Online User Manual. This user database is not related to the White Pages database. This is a separate database, which is only used for protected pages.

Instructions for setting up a category for subscription can be found in the Subscriptions chapter in the SAXOTECH Online User Manual.

- Login page
    - pbs login
    - pbs logininfo
    - pbs target
- pbs changepassword
- pbs global
- pbs subscriberdata
    - SubscriberData.pbo
- pbs userid
- Self-created users

Login page

Templates

The default template for login is **gen_login.pbs**. If you would like to have a specific template for a category, replace **gen** with the category ID. When a web guest tries to access a section that is protected, he will be directed to this page.

SAXOTECH Online needs the following tags and fields in the template.

Fields

PBS_UserID	Passes the user name to the system.
PBS_Password	Passes the password to the system.
Usefailedpage	Enables the failed page function. Normally, the web guest is returned to the login page if login fails. If this field is present with the value 1, the template gen_loginfailed.pbs will be shown instead. The field is typically used like this: `<input type=hidden name="usefailedpage" value=1>`

Example

```
<form action="<pbs:login>" method="post" name="Login" id="Login">
<input type="hidden" name="PBS_Target" value="<pbs:target>">
Username: <input type="text" name="PBS_UserID" size="14" class="input"><br>
Password: <input type="password" name="PBS_Password" size="14" class="input"><br>
<input type="submit" value="Continue &gt;&gt;" class="button">
</form>
```

pbs login

<pbs:login>

This tag must be used as an action by the template:

```
<form action="<pbs:login>" method="post" name="Login" id="Login">
```

pbs logininfo

<pbs:logininfo>

Outputs the information field from the subscription category.

pbs target

<pbs:target>

This tag is placed inside a hidden field called PBS_target:

```
<input TYPE=hidden NAME="PBS_Target"  value="<pbs:target>">
```

The system uses this field to return the web guest to the correct sectionfront or story.

pbs changepassword

<pbs:changepassword>

This tag outputs the URL to which the form must be submitted. This is also the URL for the change password form.

The user must be logged in to change her password. To access the change password page, you must use this URL:

```
/apps/pbcs.dll/nyttpassord
```

You can specify a category.

Templates

This function uses the template **gen_nyttpass.pbs**.

Form

The form is set up with the following code:

```
<FORM name="frmChangePassword" method="post" action="<pbs:changepassword>">
```

Fields

Old password	The user enters the old password in this field: `<input type="password" name="PBS_OldPassword" size=12 value="">`
New password	The user enters the new password in this field: `<input type="password" name="PBS_NewPassword1" size=12 value="">`
Repeat new password	The user re-enters the new password in this field: `<input type="password" name="PBS_NewPassword2" size=12 value="">`
Target	(Hidden field) Sets the page to which the user is sent when the password is changed: `<input type="hidden" name="PBS_Target" value= "/apps/pbcs.dll/frontpage">`
User ID	(Hidden field) Stores the user ID: `<input type="hidden" name="PBS_UserID" value= "<pbs:userid>">`

Submission form

You should use the following code to check that the passwords the user has entered are not blank and that the new password is the same in both fields:

```
<INPUT type="button" value="Continue" onClick="javascript:CheckNewPass();">
<script language="javascript">
function CheckNewPass () {
var submitForm = true;
var NewPass1 = document.frmChangePassword.PBS_NewPassword1.value;
var NewPass2 = document.frmChangePassword.PBS_NewPassword2.value;
if (NewPass1 == "" || NewPass1 == null || NewPass2 == "" || NewPass2 == null) {
submitForm = false;
alert("You must enter your new password twice to confirm that your password will be changed.");
}
if (submitForm) {document.frmChangePassword.submit();}
}
</script>
```

pbs global

<pbs:#global#>

You can also display the content of the <%@loginfailedmessage%> variable using the <pbs:#global#> tag like this:

```
<pbs:!#global# object="<%@loginfailedmessage%>">
```

This tag should be run uncached because the message can vary from user to user.

Templates

This template is used to redesign the page to which the web guest is sent when login fails and the **Usefailedpage** option is set. The template is called **gen_loginfailed.pbs**.

This page has no special tags. You can use all the tags from pbs subscriberdata. It has a special global variable that can be used in all object files used on the page.

The variable is <%@loginfailedmessage%>.

pbs subscriberdata

<pbs:subscriberdata>

This function allows you to show information about a logged-in web guest. The tags cannot show information from external subscription systems like Clickshare.

Templates

This tag can be used on all templates.

Options

Field	This section is used to specify which field should be available. Only one field can be specified.

Fields	This section is used to specify which field should be available. Multiple fields can be specified. The available fields are: • Userid - The web guest's ID. This is the ID that he uses to log on to the site. • Username - The web guest's name. • Email - The web guest's email address. • Phone - The web guest's phone number. • Fax - The web guest's fax number. • Description - A description of the web guest. • Fromdate - The date from which the account is active. • Todate - The expiration date for the account. • Password - The web guest's password. • Changepassword - This field has the value 1 if the change password flag is set. • Loggedin - This field has the value 1 if the user is logged in. This field is only available when you use the **Type** option in the tag.
Type	This option is used to specify the subscription type. This option does not have a default value. Specify the category ID for the subscription type. If you use Clickshare, the type *Clickshare* is also available.

SubscriberData.pbo

The fields you have specified in the **Field** or **Fields** sections will be available here as variables.

pbs userid

<pbs:userid>

This tag outputs the user ID for the logged in user.

Self-created users

The self-creation option lets the users create their own accounts. To use this option, you must create a template as described in the following section.

Users can be added to the database in several ways.

If you have an external database of user names and passwords, SAXOTECH Online can import these names and passwords into the database, if your system can export them. This option requires integration with your system and is done on a case-by-case basis. Please contact SAXOTECH if you want to use this option.

The SAXOTECH Online interface also allows your staff to enter users directly into the database. This is described in the *SAXOTECH Online User Manual.*

Templates

When using the self creation option you must create a template in which the users can create their accounts. This file can be named whatever you like, but SAXOTECH recommends **adduser_ovr.pbs** so it can be accessed as a sectionfront.

Fields

The following fields are mandatory. Omitting any of this code will break the page.

Date range	Set the date range in which the account is active. These dates are entered in the YYYYMMDD format.
	`<input type="hidden" name="Fra_dato" value="00000000">`
	`<input type="hidden" name="Til_dato" value="99999999">`

Email address	The user's email address. `<input type="text" name="Email" value="">`

ErrorURL	The URL to which the user will be sent if the form is filled-in incorrectly. `<input type="hidden" name="ErrorURL" value="/apps/pbcs.dll/misc?URL=/misc/error.pbs">`
Fax	The user's fax number. This field is optional. `<input type="text" name="Fax" value="">`
Form code	The template must start with the following line: `<form name="frmEditAbo" method="POST" onSubmit="return SjekkData(this)" action="/apps/links.dll/savelink?Type=6">` This code must not be changed. Changing it will break the page.
Javascript	The JavaScript in the top of the example code must be present.
Name	The user's name. `<input type="text" name="Brukernavn" value="">`
Newspaper code	Used to set the newspaper code. It must be set to your code. `<input type="hidden" name="Avis" value="GE">`

Other fields	The following fields must be present:
	`<input type="hidden" name="OrgBrukerid" value="">`
	`<input type="hidden" name="Byttpassord" value="0">`
	`<input type="hidden" name="CheckMode" value="0">`
	`<input type="hidden" name="win" value="0">`
	`<input type="hidden" name="nye" value="">`

Password	The user's password. `<input type="password" name="Password" value="">`
Phone	The user's phone number. This field is optional. `<input type="text" name="Tlf" value="">`
RedirectURL	The URL to which the user will be sent if the form has been filled-in correctly. `<input type="hidden" name="RedirectURL" value="/apps/pbcs.dll/section?category=obituaries">`

Subscribtion type	Set the category to which the subscribers should be added. The value of these fields must match the category you have created through the editing interface.
	
	`<input type="hidden" name="OrgAboType" value="SUBSCRIBE">`
	`<input type="hidden" name="Abon_type" value="SUBSCRIBE">`

User ID	The user name with which the user logs on to your site. `<input type="text" name="Brukerid" value="">`

Chapter 25 User Registration

User login can use the information in the White Pages to allow only registered users to see content or add events to the events calendar. User login is a result of user registration. Users can register to have their names in the e-mail catalog. The same database that is used for the White Pages is also used for user verification when users submit events or sign up for a newsletter.

- Advanced login
  - pbs ureg_email
  - pbs ureg_message
- Edit user registration
- Log out
- Lost password
  - PwdSent.pbo
- pbs adregusertargets
- pbs extra_income
- pbs reguserinfo
  - Userreginfo.pbo
- Setup user registration

Advanced login

The information in the White Pages can be used to allow only registered users to see content or add events to the events calendar.

To add login to a section, you must set the **Registration** option to **Advanced** on the category in the category register. The user will be presented with the login form before the she can see the page.

To force an already-logged-in user to log in again, you can use the option **&ForceLogin=1** in the link to the page.

From version 6.4.0:
To enable Facebook single-signon you must add the option **&FBLogin=1** in the link to the page. Default is 0.

Template

The template used for the login form is called **gen_userreglogin.pbs**.

Form

The following code is used to set up the form:

```
<form METHOD="POST" name="information" action="<pbs:reguser>" onSubmit="pbsSetCookie(this)">
```

Fields

Cookiename	Use this hidden field to set the name for the cookie that stores the username and password. `<input type="hidden" name="cookiename" value="usernamepassword">`

Editing info	Use the following to create a link to the page for editing the user information: `<a href="frontpage?ForceUserreg=1&Edit=1">Edit your info</a> `

Forgotten password	You can also link to a page where the user can email her password to herself.
	`<a href= "javascript:NewWindow(450,300,'/apps/pbcs.dll/misc?URL=/misc/password_email.pbs')" > Forgot your password? </a>`
	or
	`<a href="/apps/pbcs.dll/misc?URL=/misc/password_email.pbs"> Forgot your password? </a>`
	Which link you use depends on whether the page is a pop-up or not.

Link for registering	Use the following to create a link to the user registration page. `<a href="frontpage?ForceUserreg=1">Register yourself</a> `
Login	This hidden field must be in the form to tell the system that the user should log in, and not register a new user. The field is used like this: `<input type="hidden" name="Login" value="1">`
Password	The password field is called **pwd.** `<input type="Password" name="pwd" value="<pbs:#ureg_pwd#>" size="20" maxlength= "20">`
RedirectURL	Use this hidden field to send the user to a page other than the one for which you have set user login. You must add the following hidden field to the form: `<input name="RedirectURL" type="hidden" value="someURL">?`
Usecookie	You can use the field **usecookie** to allow the user to store her user name and password in a cookie so that she do not have to log in each time. `<input type="checkbox" name="usecookie" value="1">`
Username	The username field is called **e-mail** because the e-mail address is used as the username. `<input type="Text" name="email" value="<pbs:#ureg_email#>" size="20" maxlength= "40">`

Script

The following JavaScript needs to be called:

```
<script language="javascript" src="/apps/cookiesearchparams.js"></script>
```

This script contains all the cookie functions used by the form.

pbs ureg_email

<pbs:#ureg_email#>

Outputs the user's email address if she is already logged in.

pbs ureg_message

<pbs:#ureg_message#>

The system uses the function

```
<pbs:#ureg_message#>
```

to output messages from the system when the user is registering.

Edit user registration

This function allows the user to change White Pages information.

This tag is an example of how to output custom user information from the system. In this case, the information is user income. These tags will start with "extra_" followed by the field name of the desired information. This tag is used on the Edit Form.

To let the user update her information, you must add `&ForceUserReg=1&Edit=1` to the link that takes her to the White Pages page.

The link should look like this:

```
<a href="frontpage?1&ForceUserreg=1&Edit=1">Edit your info</a><br>
```

Templates

The logon form uses the template **gen_userregedit.pbs**.

The edit template is called **gen_userregeditform.pbs**. The input fields are the same as used in the registration form.

Logon form

This line sets up the form.

```
<form METHOD="POST" action="<pbs:reguser>">
```

Fields

Edit	(Hidden field) Sets the next page to be the edit form. `<input type="hidden" name= "edit" value="1">`
E-mail field	Captures the email address. `<input type="Text" name= "email" value= "<pbs:#ureg_email#>" size= "30" maxlength="40">`

Password field	Captures the password. `<input type="Password" name="pwd" value="<pbs:#ureg_pwd#>" size="20" maxlength="20">`

Edit form

After the user has logged on to the system, the form for editing her current information is shown. The form is built with the following fields.

This line sets up the form.

```
<form METHOD="POST" name="information2" action="<pbs:reguser>">
```

Fields

These fields must be present for the form to work:

<input type="hidden" name="Region" value="0000">
<input type="hidden" name="edit" value="1">
<input type="hidden" name="id" value="<pbs:#ureg_id#>">

> ⚠ The e-mail field cannot be used in the edit form.

Log out

To log out, you must use the following JavaScript to delete the cookie:

```
<script Language="Javascript">InitCookieSearchParams(document.login)</script>
<a href="javascript:EatCookie();">Log out</a><br>
<script>
function EatCookie() {
document.cookie="usernamepassword=;Path=/"
document.cookie="UserRegID=;Path=/"
document.location="/apps/pbcs.dll/frontpage?RegLogout=1"
}
</script>
```

OAMS Ad targeting eat cookie

To use the eat cookie function with OAMS ad targeting, you must add a line to the JavaScript:

```
<script Language="Javascript">
InitCookieSearchParams(document.login)
</script>
<a href="javascript:EatCookie();" class="white">Log out</a>
  |  
<a class="white" href="/apps/pbcs.dll/section?Category=REGISTER">Edit your settings</a>
<script>
function EatCookie() {
document.cookie="usernamepassword=;Path=/"
document.cookie="UserRegID=;Path=/"
document.cookie="PBCSSESSIONID=;Path=/"
document.cookie="OAMS_TARGETS_LIST=;Path=/"
document.location="/apps/pbcs.dll/frontpage?RegLogout=1"
}
</script>
```

This will ensure that the web guest is served non-targeted ads upon log out.

Lost password

This function lets the user receive her password through e-mail if it has been forgotten.

The link can look like this:

```
<a href="/apps/pbcs.dll/misc?URL=/misc/password_email.pbs">
```

If you want the form to open in a pop-up window, use this link:

```
<a href="javascript:NewWindow(450,300,
'/apps/pbcs.dll/misc?URL=/misc/password_email.pbs')">
```

Form

You set up the form with the following line:

```
<form name="emailform" action="<pbs:prog>/tips?sp=1" method="post">
```

Fields

Email address	The **To** field is used for the email address to which the password should be sent. `<input type="text" name="to" size="25">`
Email subject	Used to give the e-mail a subject line. `<input type="hidden" name="title" value="Password reminder">`
FailedNext	(Hidden field) Used to set the page to which the user will be taken if the email address is not in the database. `<input type="hidden" value="<pbs:prog>/misc?URL=/misc/password_emailfailed.pbs" name="failedNext">`
Next	(Hidden field) Used to set the page to which the user will be taken when the email has been sent. `<input type="hidden" value="<pbs:prog>/misc?URL=/misc/password_emailsent.pbs" name="Next">`

Buttons

The following code enables **Send** and **Close** buttons. You should only use a close button if you have used a pop-up window for the page.

```
<input type="submit" name="Submit" value="Submit">
<input type="button" name="Cancel" value="Cancel" onClick="javascript:window.close();">
```

PwdSent.pbo

An email is sent to the user when she requests her password. This email is sent to the email address that is provided.

The **PwdSent.pbo** must be placed in the objects folder.

<%clienthost%>	The IP-address from which the request was sent.
<%password%>	The user's password.
<%to%>	The email address.

pbs adregusertargets

<pbs:adregusertargets>

This tag is used to match user login profiles and criteria from the SAXOTECH Online database to the OAMS database for the purpose of returning ads targeted to the registered user's criteria. It is only necessary to use this tag if you decide not to use SAXOTECH Online's built-in user verification engine.

If you do use this tag, you should place it only on the first page being displayed after a user has been successfully logged in.

For more information, please see the OAMS chapter in the SAXOTECH Online User Manual.

pbs extra_income

Edit user information

This function allows the user to change White Pages information.

This tag is an example of how to output custom user information from the system. In this case, the information is user income. These tags will start with Ôextra_Ô followed by the field name of the desired information. This tag is used on the Edit Form.

To let the user update her information, you must add &ForceUserReg=1&Edit=1 to the link that takes her to the White Pages page.

The link should look like this:

```
<a href="frontpage?1&ForceUserreg=1&Edit=1">Edit your info</a><br>
```

Templates

The logon form uses the template **gen_userregedit.pbs**.

The edit template is called **gen_userregeditform.pbs**. The input fields are the same as used in the registration form.

Logon form

This line sets up the form.

```
<form METHOD="POST" action="<pbs:reguser>">
```

Fields

Edit	(Hidden field) Sets the next page to be the edit form. `<input type="hidden" name="edit" value="1">`
Email field	Captures the e-mail address.`<input type="Text" name="email" value="<pbs:#ureg_email#>" size="30" maxlength="40">`

Password field	Captures the password. `<input type="Password" name="pwd" value="<pbs:#ureg_pwd#>" size="20" maxlength="20">`

Edit form

After the user has logged on to the system, the form for editing her current information is shown. The form is built with the following fields.

This line sets up the form.

```
<form METHOD="POST" name="information2" action="<pbs:reguser>">
```

Fields

These fields must be present for the form to work:

```
<input type="hidden" name="Region" value="0000">
<input type="hidden" name="edit" value="1">
<input type="hidden" name="id" value="<pbs:#ureg_id#>">
```

All input fields available in the user registration form can be used in this form as well.

pbs reguserinfo

<pbs:reguserinfo>

When a user is logged in, you can show the user's information.

The tag must be run uncached. Otherwise, the information for the user that generated the cached file will be shown.

Templates

This function can be used on all templates.

Options

<%extra_income%>	To retrieve custom user information use `"extra_"` followed by the custom field names (for example *income*)

Userreginfo.pbo

⚠ This file can support anything contained in the [USERINFO_XXX] section of the session file as a variable

<%User_email%>	The user's email address.
<%User_ID%>	The user ID.
<%UserLevel%>	The User level for forums.

<%User_Name%>	The user name. If the user has not registered with a name, the field will contain `"None None"`
<%UserVerified%>	This field contains the value 1 if the user is a verified forum user.

Setup user registration

To have a section requiring user registration, you must set the **Registration** option to either `Simple` or `Advanced` for the category in the category register. Setting the option to `Simple` will present the user with the template **gen_userreg.pbs** before the page is shown. Setting the option to `Advanced` will present the user with the template **gen_userreglogin.pbs**.

Please see Advanced login for information about using the Advanced option for presenting a login page.

Script

The following JavaScript will open a dialog box if the user says her age is below 13.

```
<SCRIPT LANGUAGE="JavaScript">
function Checkage(theForm) {
if (theForm.age[0].selected==true) {
 alert('COPA prohibits user registration by children below the age of 13.');
 document.location.href = "/under13";
 return false;
}
}
</SCRIPT>
```

Form

The form code must be present for the form to work.

```
<form METHOD="POST" name="information2" action="<pbs:reguser>" onSubmit="return Checkage(this);"
>
```

Fields

The only fields that are required are: Email address, first name and last name. Password and repeat password are only necessary when the generate password function is not used.

It is possible to add custom fields to your users. These fields are not available through the edit interface but they are available on the templates. See the options for the pbs reguserinfo tag.

The fields are added to the template as follows:

<input type=text name="extra_age" value=<pbs:#extra_age#>>

In the example above, a new field called age will be stored with the information written in the input form. The extra field's name can be used for target advertising in OAMS. So, if you create target advertising, then the names of the fields in OAMS and the extra fields from the user registration will make target advertising work.

Address 1	The first address line. `<input type="text" name="address" size="15" value="<pbs:#ureg_address#>">`
Address 2	The second address line. `<input type="text" name="address2" size="15" value="<pbs:#ureg_address2#>">`

Age	Verifies that the user is over the age of 13. This is done to comply with the United States Child Online Protection Act.
	`<SELECT NAME="age" SIZE=1>`
	`<OPTION VALUE="13 or under" SELECTED>under 13</OPTION>`
	`<OPTION VALUE="13over">over 13</OPTION>`
	`</select>`

Cellphone	The cell phone number. `<input type="text" name="cellphone" size="12" value="<pbs:#ureg_cellphone#>">`
Cookiename	(Hidden field) Sets the cookie name that is used to store the username and password `<input type="hidden" name="cookiename" value="usernamepassword">`
City	The city. `<input type="text" name="city" size="15" value="<pbs:#ureg_city#>">`
Date of birth	The user's date of birth. `<input type="text" name="dob" size="8" value="<pbs:#ureg_dob#>">`

Email format	Selects the email format.
	`<select name="mailformat">`
	`<option value="0">Select format >>`
	`<option value="1" <pbs:#ureg_mailformat1#>>Text`
	`<option value="2" <pbs:#ureg_mailformat2#>>HTML`
	`</select>`

Email address	The user's email address. It is used as user name. `<input type="Text" name="email" value="<pbs:#ureg_email#>" size=30>`
Email subject	(Hidden field) Sets the email subject. `<input type="hidden" name="emailsubject" value="User registration">`

Email text	Creates the email body. The field can have multiple lines. All fields in the registration form can be used here.
	`<input type="hidden" name="emailtext"`
	`value="You are now registered at widgets.com with the following information:`
	`Email: #ureg_email#`
	`Password: #ureg_pwd#`
	`Other fields go here...">`

First name	The user's first name. `<input type="Text" name="fname" value="<pbs:#ureg_fname#>" size="30" maxlength="40">`
Gender	This field allows the user to pick a gender. `<select name="sex"> <option value="M" <pbs:#ureg_sex_male#>> Male <option value="F" <pbs:#ureg_sex_female#>> Female </select>`
Generate password	Use this hidden field if the system should generate a password for the user. When this option is used, the fields enabling the user choose a password must not be present. The email option must be enabled so that the password can be mailed to the user. `<input type="hidden" name="generatepassword" value="1">`
Home phone	The home phone number. `<input type="text" name="homephone" size="12" value="<pbs:#ureg_homephone#>">`
Homepage	The user's homepage. `<input type="text" name="homepage" size="15" value="<pbs:#ureg_homepage#>">`
ID	(Hidden field) This field must also be present to keep track of the ID code. `<input type="hidden" name="id" value="<pbs:#ureg_id#>">`
Last name	The user's last name. `<input type="Text" name="lname" value="<pbs:#ureg_lname#>" size="30" maxlength="40">`
Middle name/Initial	The user's middle name or initial. `<input type="Text" name="mname" value="<pbs:#ureg_mname#>" size="30" maxlength="40">`
Password	The user's password. `<input type="Password" name="pwd" value="<pbs:#ureg_pwd#>" size=20>`

Phone number	The user's phone number. `<input type="Text" name="phone" value="<pbs:#ureg_phone#>" size="15" maxlength="40">`
Redirect URL	(Hidden field) Sets the redirect URL. This field should only be used when the server generates passwords. It directs the user to the page informing her that a password has been sent to her e-mail address. `<input type="hidden" name="RedirectURL" value="/">`
Region	(Hidden field) This field must be present. `<input type="hidden" name="Region" value="0000">`
Repeat password	This field is used to repeat the password. This is done so the user can verify that the password is typed correctly. `<input type="Password" name="pwdr" value="<pbs:#ureg_pwdr#>" size="20" maxlength="20">`
Send email	(Hidden field) Sends an email with the user information. `<input type="hidden" name="sendemail" value="1">`
Sender's email	(Hidden field) Sets the sender's email address. `<input type="hidden" name="fromaddress" value="someaddress@goupstate.com>`
Sender name	(Hidden field) Sets the sender's name. `<input type="hidden" name="fromname" value="GoUpstate.com">`
Show in directory	This checkbox allows the user to determine whether she will be shown in the White Pages. `<input type="checkbox" name="show" <pbs:#ureg_show#> value="1">`
State/Country	The country or state. `<input type="text" name="country" size="15" value="<pbs:#ureg_country#>">`
Submit	This button submits the information to the server. `<input type="Submit" name="Submit" value=" Register ">`

Tip	This field is used for a newsletter. The newsletter is unavailable at the moment. `<input type="checkbox" name="tip" <pbs:#ureg_tip#>>`

Tip interval	Sets the tip interval for the newsletter. This a separate newsletter function separate from the headlines newsletter. This function is currently not available.
	`<select name="mailinterval">`
	`<option value="0">Select interval >>`
	`<option value="1" <pbs:#ureg_mailinterval1#>>Daily`
	`<option value="7" <pbs:#ureg_mailinterval7#>>Weekly`
	`</select>`

Use cookie	This checkbox allows the user to store the user name and password in a cookie so that she does not have to type the user name and password each time a protected section is accessed. `<input type="checkbox" name="usecookie">`
White pages	This checkbox controls whether the user is willing to have her name in your email catalog. `<input type="Checkbox" name="show" value="1" <pbs:#ureg_show#>>`
Zipcode	The zip code `<input type="text" name="zip" size="5" value="<pbs:#ureg_zip#>"">`

Data retrieval fields

These tags are used to pass information back to the form if the user fills in a field incorrectly must complete the form again. These fields save the user from retyping information.

There is a matching tag for each field in the form. For example:

To retrieve the information for the field email, use the tag:

```
<pbs:#ureg_email#>
```

This field is entered in the value field for the input field with the name email like this:

```
<input type="Text" name="email" value="<pbs:#ureg_email#>" size="30" maxlength="40">
```

Chapter 26 Staff Listing

Staff listing is used to build employee white pages. This function is usually used to display contact information.

- pbs departmentlist
  - DepList.pbo
  - DepListItem.pbo
- pbs departmentoverview
  - DepOverview.pbo
  - DepOverviewGroupHeader.pbo
  - DepOverviewHeader.pbo

- DepOverviewItem.pbo
- pbs staffinfo
  - StaffInfo.pbo

pbs departmentlist

<pbs:departmentlist>

This function lists all departments registered in the system. This tag can be placed on all templates.

Options

Location	Used to filter the list based on the information in the **Office** field. For more information please see DepOverviewItem.pbo .By default the tag lists all locations. Enter `location="office"` to filter the list so that only records where the office matches will be shown. When you click on this link the resulting list will be filtered as well.
Objectclass	Sets the object class.

DepList.pbo

This file is optional.

<%DepListItems%>	This field lists the items.

DepListItem.pbo

This file is output for each department.

<%Department%>	This field contains the name of the department.
<%Link%>	This field contains the link to the staff list.

pbs departmentoverview

<pbs:departmentoverview>

The tag for displaying the staff list can be placed on any template. The usual way to do it is to place the tag on an overview page like **contact_ovr.pbs.**

Options

City	This option is used to filter based on the city field. By default, all cities are listed. Enter `city="xxx"` to limit the list to those staff members with **xxx** in the city field.
Department	This option is used to filter based on the department field. Note: **Department** is case-sensitive. • Correct: `<pbs:departmentoverview department="Markedsavdelingen">` • Incorrect: `<pbs:departmentoverview Department="Markedsavdelingen">`

Location	This option is used to filter based on the location field. By default all locations are listed. This value is entered into the office field in the user interface. Enter `location="office"` to filter the list to a single office. When this option is used on your pbs tag you can use the sort field in your edit interface.
Objectclass	Sets the object class.
Sortby	This option makes it possible to sort the staff list by first name, work phone, occupation and more. This is done in the following way: `<pbs:departmentoverview sortby="firstname">` Without the **sortby** parameter specified, the list sorts by last name by default. The following options can be used (case insensitive): • WORKTLF - work phone • STILLING or OCCUPATION • PRIVTLF - private phone • MOBILTLF - cell phone • POSTNR or ZIP • POSTSTED or CITY

DepOverview.pbo

<%DepOverview%>	This field lists the items.

DepOverviewGroupHeader.pbo

Output for each group in the department.

<%DepGroupName%>	The name of the group.
<%DepOverviewItems%>	Outputs all employees in the group.

DepOverviewHeader.pbo

Outputs a header for each department.

<%DepEmail%>	The group e-mail address for the department.
<%DepFax%>	Outputs the department fax number.
<%DepName%>	The name of the department.
<%DepOverviewGroups%>	Outputs the department groups containing all employees and un-grouped employees.
<%DepPhone%>	Outputs the department phone number.

DepOverviewItem.pbo

Output for each employee.

<%Address1%>	This field displays address line 1.
<%Address2%>	This field displays address line 2.
<%Address3%>	This field displays address line 3.
<%Address4%>	This field displays address line 4.
<%City%>	This field contains the city name.

<%DepFax%>	The department fax number.
<%DepGroupName%>	The name of the group.
<%DepName%>	The name of the department.
<%DepPhone%>	The department phone number.
<%\Email%>	The employee's email address.
<%EmailHome%>	This field contains the personal email address for the employee.
<%Firstname%>	The employee's first name.
<%ImgLink%>	The source address for the image associated with the employee. The maximum size for staff images is 500 x 500 pixels, which is the default. To limit the image to a smaller size, determine the maximum width as in the example below, which limits the width to 100 pixels: `<img src="<%ImgLink%>&MaxW=100" border="0">`
<%Info%>	This field contains the information field.
<%JobTitle%>	The job title.
<%Name%>	The employee's name in the format: Last name, first name.
<%Office%>	This field is used for office location.
<%PhOffice%>	Office phone number.
<%PhDirect%>	Direct phone number
<%PhHome%>	Home phone number.
<%PhCellular%>	Cell phone number.
<%PhBeeper%>	Pager number.
<%Phfax%>	This field contains the fax number.
<%Residence%>	This field is used for home location. Typically community/county name.
<%Staffid%>	The ID for the employee. This field is used to provide a link to the details page for the person. It can be used like this: `<a href="/apps/pbcs.dll/personalia?ID=<%StaffID%>"><%Name%></a>`
<%Surname%>	The employee's last name.
<%WebHome%>	This field contains a link to a personal webpage.
<%WebOffice%>	This field contains a link to a work-related webpage.
<%Zip%>	This field contains the Zip/Postal code.

pbs staffinfo

<pbs:staffinfo>

This tag is used to show the details for one person. The tag must be placed on the template **gen_pers.pbs**.

Options

Objectclass	Sets the object class.

StaffInfo.pbo

This file uses the same fields as DepOverviewItem.pbo, plus the following fields:

<%Firstname%>	The employee's first name.
<%Surname%>	The employee's last name.

Chapter 27 Tip Functions

The tip functions allow web guests to send links to other internet users. This can include links to stories, classified ads or guide entries in the system.

- External tip page
- pbs article-tiplink
  - ArtTips.pbo
  - ArtTipsText.pbo
  - ArtTipsTextHtml.pbo
  - ClassTipsText.pbo
  - ClassTipsTextHTML.pbo
  - GuideTipsText.pbo
  - GuideTipsTextHTML.pbo
  - Tips.pbo
  - TipsText.pbo
- Setup tip functions

External tip page

This tip page is used by the Story, Classifieds and Guide components. The template is called **gen_tip.pbs**.

This table describes what the lines in the form do. Unless stated, all lines are mandatory.

`<form action="<pbs:#action#>" method="POST" name="Tipform">`	Starts the form. The action must be `<pbs:#action#>`.
`<pbs:#hidden#>`	Outputs the hidden system fields.
`<input type="hidden" value= "<pbs:prog>/misc?URL=/misc/tip_emailsent.pbs&NoMal=1" name= "Next">`	Sets the page to which the web guest will be directed after the email has been sent.
`<script>document.write(document.Tipform.tittel.value);</script>`	Writes the category name for the section being viewed. Optional.
`<input type="text" name="from" size=20>`	Captures the user's name. Optional, but should be present.
`<input type="text" name="fromemail" size=20>`	Captures the user's email address. Optional, but should be present.
`<input type="text" name="to" size=20>`	Captures the recipient's email address.
`<textarea name="message"cols=20 rows=3></textarea>`	Outputs the message area that is used for a message to the recipient. Optional, but should be present.

`<input type="submit" name="Submit" value="Send">`	Outputs the **Submit** button.
`<input type="button" name="Cancel" value="Cancel" onClick= "javascript:window.close();">`	Outputs a **Close** button. Optional. This feature should only be used if you open the tip form in a separate window.
`</form>`	Closes the form.

For an article tip, please see ArtTipsText.pbo. For a Classifieds tip, please see ClassTipsText.pbo.

pbs article-tiplink

<pbs:article-tiplink>

This tag creates a link to the *Tip a friend* page. There are two versions of the article tip. The form integrated into the page and the pop-up page version. This section covers the integrated tip form. For information about the pop-up version, see External tip page.

The field <%tips%> in ArtTips.pbo triggers the form being displayed. The tip page is described in the chapter External tip page. This tag can be used on all article templates. This tag has no options. The function requires that the **Tips active** checkbox has been marked on the properties page for the category.

Example code
This piece of code will open a new window when a web guest clicks the text *Email this to a friend*.

```
<SCRIPT LANGUAGE="JavaScript1.2">
function NewWindow(height,width,url) {window.open(url,"ShowProdWindow",
"menubars=0,scrollbars=1,resizable=1,height="+height+",width="+width);
}
</SCRIPT>
<a href="javascript:NewWindow(450,300,'<pbs:article-tiplink>');">Email this to a friend</a>
```

Check field values 1

To check that the **To** field has been filled in, use the following code:

```
onSubmit="javascript:if (this.To.value=='') {alert('You must fill in the recipient address!');
return false;} "
```

Check field values 2

This function is used to check multiple fields and it is included in the example:

```
<script language="JavaScript">
function checkTipsForm(f) {
 if ((f.To.value=='') || (f.Fromemail.value=='') || (f.From.value=='') || (f.Message.value==''))
{
alert('All fields must be filled in!');
return false;
 }
}
</script>
<form action="<%Action%>" method="POST" onSubmit="javascript:return checkTipsForm(this);">
```

ArtTips.pbo

<%Action%>	Outputs the correct URL for the action of the form. it must be placed inside the form tag like this: `<form action="<%Action%>" method="POST">`
<%from%>	The sender's name.
<%Fromemail%>	The sender's email address.
<%Html%>	If this field has the value 1, the file ArtTipsTextHtml.pbo will be used instead of the normal file. This field should be used as a hidden field.
<%ImgLink%>	The link to the story's picture.
<%Message%>	This text area is used for the message entered by the web guest.
<%Next%>	This hidden field contains the information needed to redirect the web guest to a page after the tip has been sent. To redirect back to the story, use the following code: `<input type="hidden" value="<pbs:link artikkel inherit href>" name="Next">`
<pbs:#hidden#>	Outputs the hidden fields needed by the system.
<%\Title%>	This field contains the title of the story and must be placed in a hidden field named "Tittel" like this: `<input name="Tittel" type="hidden" value="<%\Title%>">`
<%To%>	The recipient's email address.

ArtTipsText.pbo

This file is the e-mail that is sent to the tip recipient.

<%from%>	The name entered by the sender.
<%Fromemail%>	The email address provided by the sender.
<%Message%>	The message from the sender.
<%Link%>	The link to the story.
<%\Title%>	The story's title.
<%To%>	The recipient's email address.

ArtTipsTextHtml.pbo

This file has the same fields as the normal file. It is used when the flag HTML is set to 1 in the object file.

ClassTipsText.pbo

The tip function allows a web guest to send an email to someone that might be interested in the ad he is viewing. This function is also used to send tips about stories.
The field

```
*<%TipLink%>*
```

in ClassifiedsItem.pbo contains the link to the tip page. The external tip page is used to send the tip.

This file is used for the email sent by the classifieds tip. This file has the same fields as ArtTipsText.pbo.

ClassTipsTextHTML.pbo

The tip function allows a web guest to send an email to someone that might be interested in the ad he is viewing. This function is also used to send tips about stories.
The field

```
<%TipLink%>
```

in ClassifiedsItem.pbo contains the link to the tip page. The external tip page is used to send the tip.

This file has the same fields as ClassTipsText.pbo. It is used when the flag HTML is set to 1 in the object file.

GuideTipsText.pbo

The tip function is also available for guide entries. The field

```
*<%TipLink%>*
```

contains the link to the external tip page.

The email sent to the web guest uses GuideTipsText.pbo. It has the same fields as ArtTipsText.pbo.

GuideTipsTextHTML.pbo

The tip function is also available for guide entries. The field

```
*<%TipLink%>*
```

contains the link to the external tip page.

This file has the same fields as the GuideTipsText.pbo. It is used when the flag HTML is set to 1 in the object file.

Tips.pbo

<%Action%>	Outputs the correct URL for the action of the form. it must be placed inside the form tag like this: `<form action="<%Action%>" method="POST">`
<%from%>	The sender's name.
<%Fromemail%>	The sender's email address.
<%Html%>	If this field has the value 1, the file ArtTipsTextHtml.pbo will be used instead of the normal file. This field should be used as a hidden field.
<%ImgLink%>	The link to the story's picture.
<%Message%>	This text area is used for the message entered by the web guest.
<%Next%>	This hidden field contains the information needed to redirect the web guest to a page after the tip has been sent. To redirect back to the story, use the following code: `<input type="hidden" value="<pbs:link artikkel inherit href>" name="Next">`
<pbs:#hidden#>	Outputs the hidden fields needed by the system.
<%\Title%>	This field contains the title of the story and must be placed in a hidden field named "Tittel" like this: `<input name="Tittel" type="hidden" value="<%\Title%>">`
<%To%>	The recipient's email address.

TipsText.pbo

The TipsText.pbo file is used when no function-specific file exists. This file has the same fields as ArtTipsText.pbo.

Setup tip functions

The **TipsText.pbo** file is used when no function-specific file exists. This file has the same fields as ArtTipsText.pbo. If this file does not exist, the system will send an email based on system defaults. It is likely that the default message is in Norwegian.

In order to use tip functions, the following settings must be added to the GENERAL section of the Publicus.ini:

MailID
MailSender
MailServer

Chapter 28 Images

This chapter contains general information about images in the system and information about the permanent gallery. Please note that SAXOTECH Online has a permanent gallery and a story based gallery. The two galleries are accessed at different locations in the editing interface.

- Access control for images
  - PhotoGalleryExtLogin.pbo
- Story-linked photo gallery
- Generate graphical text
- Image settings
- Order picture
- pbs link and images
- pbs photogallery
  - ArticleConnectionItem.pbo and images
  - ArticleConnections.pbo and images
  - PhotoGallery.pbo
- pbs photogallery-latest
  - PhotoGalleryLatest.pbo
  - PhotoGalleryLatestItem.pbo
- pbs photogallery-thumbnails
  - PhotoGalleryThumbnailItem.pbo
  - PhotoGalleryThumbnails.pbo
- pbs ptshowimage
  - ShowImage.pbo
- Photo album export
  - ExpPhotoAlbum.pbo
  - ExpPhotoAlbumItem.pbo
  - ExpPhotoAlbums.pbo
- Set picture as desktop background
- Watermark images
- pbs ptimages
  - ImageItem.pbo
  - ImageItemRowEnd.pbo
  - ImageItemRowStart.pbo
- PhotoGalleryItemExtraFieldItem.pbo
- PhotoGalleryItemExtraFields.pbo
- PhotoGalleryLatestItemExtraFieldItem.pbo
- PhotoGalleryLatestItemExtraFields.pbo
- PhotoGalleryLatestItemTaxonomy.pbo
- PhotoGalleryLatestTaxonomyItem.pbo
- PhotoGalleryTaxonomyItem.pbo

Access control for images

It is possible to control the access to images. In order to give or limit access to certain galleries, you can use Access Control. For more information on Access Control, please see the *SAXOTECH Online User Manual.*

To set up Access Control in the ini file, please see EXTLOGIN section#AccessControl.

Template

Access Control uses the template **gen_gallery_extlogin.pbs**.

PhotoGalleryExtLogin.pbo

<%StandardText%>	The text to be output. For example it could say "You do not have access to this gallery, but you can purchase access, by clicking..."

Story-linked photo gallery

The photo gallery connected to a story is usually used when you have too many photos for the story. For example, you have a travel story from an exotic location and you would like to show 20 photos. How to use the story-linked photo gallery is described in the *SAXOTECH Online User Manual.*

Linking to a photo gallery is documented in the section pbs photogallery-links.

Generate graphical text

The **Pbcsi.dll**, also known as the Image dll, has the ability to generate GIF images containing a text string. This will allow you to build graphical headers for your pages without having to edit the images directly.

Tag

This function has no tag. You pass parameters to the **image dll**. Example:

```
<img src="/apps/pbcsi.dll/text?
text=Test&Fontname=Arial&Fontsize=16&Width=300&Height=-1
&FontColor=FFFFFF&BGColor=000000&XOffset=8&YOffset=0">
```

Template

This function can be used on all pages.

Options

BGColor	Specifies the background color. The color is set using an RGB triplet.
Bold	Makes the text bold. `Enter bold=1` to enable this function.
Fontcolor	Specifies the font color. The color is set using an RGB triplet.
Fontname	Specifies the font. Only fonts installed on the image server can be used.
Fontsize	Specifies the font size.
Height	Specifies the height of the image. Setting this value to `-1` will make the system scale the image.
Italic	Makes the text italic. Enter `italic=1` to enable this function.
Text	Sets the text string.

Transparent	Makes the background transparent. Enter `transparent=1` to enable this function.
Underline	Makes the text underlined. Enter `underline=1` to enable this function.
Width	Specifies the width of the image. Setting this value to `-1` will make the system scale the image.
Xoffset	Specifies the padding in pixels on the left hand side of the text string.
Yoffset	Specifies the padding in pixels on top of the text string.

Example code

This document will allow you to experiment with the image functionality:

```
<HTML>
<BODY BGCOLOR=#EEEEEE>
<SCRIPT Language="Javascript">
<!--
function Update() {
var frm=document.Settings;
var img=document.getElementById("image");
var
url='/apps/pbcsi.dll/text?text='+escape(frm.Text.value)+'&Fontname='+escape(frm.Font.value)+'&Font
if (frm.Bold.checked) {url = url + '&Bold=1';}
if (frm.Italic.checked) {url = url + '&Italic=1';}
if (frm.Underline.checked) {url = url + '&Underline=1';}
if (frm.Transparent.checked) {url = url + '&Transparent=1';}
img.src = url;
}
-->
</SCRIPT>
<FORM NAME="Settings">
<TABLE Border=0>
<TR>
<TD>Width</TD><TD><INPUT Type="Text" Name="Width" Size=5 value="300"></TD>
</TR>
<TR>
<TD>Height</TD><TD><INPUT Type="Text" Name="Height" Size=5 value="-1"></TD>
</TR>
<TR>
<TD>Text</TD><TD><INPUT Type="Text" Name="Text" Size=40 value="Test"></TD>
</TR>
<TR>
<TD>Fontname</TD><TD><INPUT Type="Text" Name="Font" Size=40 value="Arial"></TD>
</TR>
<TR>
<TD>Fontsize</TD><TD><SELECT Name="Fontsize"><OPTION Value="-1">Automatic<OPTION Value="10"
>10pt<OPTION Value="12">12pt<OPTION Value="16">16pt<OPTION Value="20">20pt<OPTION Value="24"
>24pt<OPTION Value="30">30pt<OPTION Value="36">36pt</SELECT></TD>
</TR>
<TR>
<TD>Fontcolor</TD><TD><INPUT Type="Text" Name="Fontcolor" Size=8 value="FFFFFF"></TD>
</TR>
<TR>
<TD>Backgroundcolor</TD><TD><INPUT Type="Text" Name="BGColor" Size=8 value="000000"></TD>
</TR>
<TR>
<TD>Text X-Offset</TD><TD><INPUT Type="Text" Name="XOffset" Size=5 value="0"></TD>
</TR>
<TR>
<TD>Text Y-Offset</TD><TD><INPUT Type="Text" Name="YOffset" Size=5 value="0"></TD>
</TR>
<TR>
<TD COLSPAN=2>Bold<INPUT TYPE=Checkbox Name=Bold> Italic<INPUT TYPE=Checkbox
Name=Italic> Underline<INPUT TYPE=Checkbox Name=Underline> Transparent<INPUT
TYPE=Checkbox Name=Transparent> </TD>
</TR>
<TR>
<TD COLSPAN=2><hr></TD>
</TR>
<TR>
<TD COLSPAN=2 ALIGN=Right><INPUT Type="Button" Value="Update" onClick="Update()"></TD>
</TR>
</TABLE>
</FORM>
<img id=image src="/g/t.gif">
</BODY>
</HTML>
```

Image settings

These settings are used in the URL of the **Pbsci.dll**, also known as the Image dll, to control the way the image is shown.

Background color	This option sets the background color when you use the rounded corner and drop shadow function. To set the option, enter
	`&BgColor=AABBCC`
	where
	`AABBCC`
	is the hex triplet for the color. Example:
	`<img src="<%image%>&RcRadius=20&BgColor=90AAAA">`

Border color	Sets the border color. The default color for the border is black. Add the following option to change the border color:
	`&BColor=RRGGBB`
	The color is set using the standard HTML color triplet. This code will make the border red:
	`<img src="<%image%>&BColor=FF0000" border=0>`

Border on/off	Add the following text to the link to enable the border:
	`&Border=1`
	Border is enabled by default. Enter
	`border=0`
	to disable the border. Example:
	`<img src="<%image%>&border=0" border=0>`

Drop shadow	This function is used to give photos a drop shadow. To set the option, &DsRadius=XX, where XX is the number of pixels in the offset. It is recommended that this option should not be used simultaneously with the rounded corners option. Example: `<img src="<%image%>&DsRadius=20">`

Image height	The image height is usually set in the tag that is used for displaying the picture. The tag automatically ads this option. It is also possible to code the height in the object file. This is usually done when you use article classes. By default, the height is set by the tag. Enter
	`&Maxh=X`
	to set the height of the picture to
	`X`
	pixels. Example:
	`<img src="<%image%>&Maxh=80">`

Image quality	You can override the value for the JPEG compression level. For most pictures, the quality is at the right level, but for larger pictures you may want to increase the quality. The default quality level is set to 60.To change this value, add
	`&Q=X`
	to the image URL, where
	`X`
	is the quality level. The option is used like this:n
	`<img src="<%image%>&Q=80">`
	1 is the lowest quality and 100 is the highest.

Image width	The image width is usually set in the tag that is used to display the picture. The tag automatically ads this option. It is also possible to code the width in the object file. This is usually done when you use article classes. By default, the width is set by the tag.Enter
	`&Maxw=X`
	to set the width of the picture to
	`X`
	pixels. Example:
	`<img src="<%image%>&MaxW=580">`

Rounded corners	This function makes the corners of the image round. To set the radius of the corner, you specify the option
	`&RCRadius=X`
	where
	`X`
	is the number of pixels. It is recommended that this option should not be used simultaneously with the drop shadow option. Example:
	`<img src="<%image%>&RcRadius=20">`

Order picture

If you would like to have an order form in which web guests can purchase prints of pictures, include the following link in your ArticlePicture.pbo:

```
<a href="<%pbcs.dll%>/misc?url=/misc/orderpicture.pbs&<%ImgKeyParams%>">
Order Picture
</a>
```

Template

In this example, the file **orderpicture.pbs** in the /misc folder is used.

Form

The form is setup with the following line:

```
<FORM METHOD="POST" ACTION="/apps/xsendmail.dll">
```

Hidden fields

These fields are used to set options.

Article URL	Sets the URL to the story that is linked to the photo ordered by the web guest. You must replace www.yoursite.com with the address to your website. `<INPUT TYPE=hidden NAME="Article URL" VALUE="<a href='http://www.yoursite.com<pbs:!link artikkel inherit href>'>Click here</a>">`
From	Sets the from address for the email. `<INPUT TYPE=hidden NAME="FROM" VALUE="Webserver <webserver@publicus.com>">`
Lang	Sets the language of system messages. `<INPUT TYPE=hidden NAME="LANG" VALUE="ENG">`
Photo URL	Sets the URL to the photo the web guest has ordered. You must replace www.yoursite.com with the address to your website. `<INPUT TYPE=hidden NAME="Photo URL" VALUE="<a href='http://www.yoursite.com<pbs:!link inherit href text=img:ondate=0>'>Click here</a>">`
Redirect URL	Sets the URL to which the web guest will be sent after the form is filled out correctly. `<INPUT TYPE=hidden NAME="REDIRECT_URL" VALUE="/apps/pbcs.dll/misc?url=/misc/orderpicture_ok.pbs">`
Required	Determines which fields in the form are required. `<INPUT TYPE=hidden NAME="REQUIRED" VALUE="Name,Address,CityStateZip,Email">`
Subject	Sets the subject of the email sent to your staff. `<INPUT TYPE=hidden NAME="SUBJECT" VALUE="Order a picture.">`

Title	Sets the title of the table that contains the information in the email. `<INPUT TYPE=hidden NAME="TITLE" VALUE="Order a picture">`
To	Sets the recipient of the e-mail. This is the e-mail address of the person handling picture orders.`<INPUT TYPE=hidden NAME="TO" VALUE="eh@saxotech.no">`

Input fields

Include the required input fields, like name, address and phone number. The fields are defined in the following way:

```
<input type="text" name="Name" size="20" maxlength="40">
```

Replace

```
"Name"
```

with the name of the field.

Buttons

You must have a **Submit** button in the form.

Thank you page

This template must be called **orderpicture_ok.pbs** and be put in the /**misc** folder if you are using the paths in this example. This page can contain whatever you like.

pbs link and images

<pbs:link>

This function is used to give the web guest the ability to zoom in on a picture in a story. This function uses the `<pbs:link>` tag. You must use the tag like this:

```
<img src="<pbs:!link inherit ondate=0 href text=img:>&MaxW=580&title=1">
```

Template
The template used to zoom can be called whatever you like. You must specify the desired template in the link.

Link to zoomed picture
To link to the template used for the zoomed picture, you must place the following link in the object file with picture:

```
<a href="/apps/pbcs.dll/misc?url=/
```

```
*misc/zoom.pbs*
```

```
&<%ImgKeyParams%>">Zoom</a>
```

It is usually used in a pop-up window. If you would like to use this function in a pop-up window, you must use the following JavaScript:

```
<a href=
"javascript:NewWindow(460,300,'/apps/pbcs.dll/misc?url=/misc/zoom.pbs&<%ImgKeyParams%>');"
>Zoom</a>
```

Options

Logo	Please see images_Watermark_Images#Image_Watermark for more information.
Title	Please see images_Watermark_Images#Text_Watermark for more information.

Width	You can set the width of the picture by appending
	`&Maxw=X`
	where
	`X`
	is the width of the image, in pixels.

pbs photogallery

<pbs:photogallery>

Options

ObjectClass	Sets the object class.
ShowExtraFields **From version 6.2.0**	Use to show extra fields - default is 0. `showextrafields=1`
ShowTaxonomy **From version 6.2.0**	Use to show taxonomy words - default is 0. `showtaxonomy=1`
IgnoreStatus **From version 6.2.0**	Use if publish status is to be ignored - default is 0. `ignorestatus=1` Using it will display published and unpublished galleries.

The template for the photo gallery is called **gen_gallery.pbs**. It must be placed in the template folder. It is possible to create templates for each category. Link to the gallery page from a story, or by using the pbs photogallery-latest tag.

If you want to link to the gallery manually, you must specify all parameters like this:

```
<a href="apps/pbcs.dll/gallery?Date=20030609&Category=REPOSITORY&id=609001&Ref=PH">Link</a>
```

This tag can also be used to display a zoomed version of the photo in the gallery. This function uses the template **gen_galleryzoom.pbs**. This function uses the same options and object files below.

ArticleConnectionItem.pbo and images

This file is output for each story that is linked to the photo gallery.

<%Heading%>	The title of the story.
<%Link%>	The link to the story.

ArticleConnections.pbo and images

Used if links to the connected stories should be output.

<%ConnectionItems%>	Outputs each link.

PhotoGallery.pbo

<%ArticleConnectionItems%>	Outputs links to all stories that are connected to the photo gallery.
<%Byline%>	The byline for the photo.
<%Caption%>	The caption for the photo.
<%CurrentPicture%>	The number of the picture being viewed.
<%FirstURL%>	A link to the first photo in the album, except when viewing the first photo. It is usually used like this: `[%{FirstURL} <a href="<%FirstURL%>">First</a> %|% First %]`
<%GalleryText%>	The gallery description.
<%GalleryTitle%>	The title for the gallery.
<%GalleryZoomLink%>	A link to the same photo on a different template. The field is described in pbs link.
<%Height%>	The height of the picture.
<%ImgParams%>	This field contains all parameters for the image. The field can be used to link to an order form for the image.

<%JumpMenuOptions%>	Builds a drop-down menu, which enables web guests to jump to any picture in the gallery. It is built like this:
	`<form>`
	`Jump to:`
	`<select`
	`onChange="`
	`document.location.href=this.options[this.selectedIndex].value;`
	`this.options[this.selectedIndex].text='`
	`...Wait...';">`
	`<%JumpMenuOptions%>`
	`</select>`
	`</form>`
	When you use this option, it is important that the `<select>` tag is written on one line

<%LastURL%>	A link to the last photo in the album, except viewing the last photo. It is usually used like this `[%{LastURL} <a href="<%LastURL%>">Last</a> %|% Last %]`
<%NextURL%>	A link to the next photo in the album, except when viewing the last photo. It is usually used like this: `[%{NextURL} <a href="<%NextURL%>">Next</a> %|% Next %]`
<%PictureCount%>	The total number of photos in the gallery.
<%PictureURL%>	The URL for the picture. This field is usually used in the following piece of code: `<img src="<%PictureURL%>&Maxw=400&Maxh=400">`

<%PreviousURL%>	A link to the previous photo in the album, except when viewing the first photo. It is usually used like this: `[%{PreviousURL} <a href="<%PreviousURL%>">Previous</a> %|% Previous %]`
<%\Title%>	The title of the picture.
<%Width%>	The width of the picture.
<%UserSubmitDeleteLink%> **From version 6.2.0**	URL that if visited will delete the story - in case the story is a user submitted story and the user is the creator.
<%GalleryID%> **From version 6.2.0**	Contains the lopenr / StoryID of the gallery.
<%GalleryCategory%> **From version 6.2.0**	Same as Category.
<%GalleryDate%> **From version 6.2.0**	Contains the date of the gallery.
<%Location%> **From version 6.2.0**	Contains the location field value.
<%PhotoGalleryStatus%> **From version 6.2.0**	Contains the publish status of the photo gallery. It can be published (=1) or unpublished (=0). This is only returned if the photogallery-latest tag has the ignorestatus=1 parameter.
<%CreatedByUserID%> **From version 6.2.0**	Contains the creator's user ID if photo gallery is a blog entry.
<%CreatedByFirstName%> **From version 6.2.0**	Contains the creator's first name if photo gallery is a blog entry.
<%CreatedBySurName%> **From version 6.2.0**	Contains the creator's last name if photo gallery is a blog entry.
<%CreatedByEmailAdr%> **From version 6.2.0**	Contains the creator's email address if photo gallery is a blog entry.
<%BlogEntry%> **From version 6.2.0**	Whether the photo gallery is user generated (=1) or not (=0).
<%ISOCreatedDate%> **From version 6.2.0**	ISO formatted created date.
<%ISOModifiedDate%> **From version 6.2.0**	ISO formatted modified date.
<%Taxonomy%> **From version 6.2.0**	Contains the taxonomy words of the photo gallery. See PhotoGalleryLatestItemTaxonomy.pbo
<%ExtraFields%> **From version 6.2.0**	Contains the extra fields of the photo gallery. see PhotoGalleryLatestItemExtraFields.pbo
<%CategoryID%> **From version 6.2.0**	Contains the category ID of the photo gallery.
<%CategoryDescription%> **From version 6.2.0**	Contains the category name.
<%AvgRanking%> **From version 6.4.0**	Contains the average ranking of a Photo.
<%RankingLink%> **From version 6.4.0**	The RankingLink does not contain the actual ranking value this must be added to the link by using client side scripting. The Following must be added to the link before submitting the link to the DLL '<%RankingLink%>' + '&rating=' + y Where y is the ranking given by a user.
<%TotalNoofVotes%> **From version 6.4.0**	Contains the total number of votes on the Photo.

pbs photogallery-latest

This function output links to the newest photo galleries in the system. The tag can be used on all templates.

Options

Category	By default, photo galleries for all categories are shown. Enter `category="CATEGORY"` to show galleries for the defined category.
Count	By default, the 5 latest galleries are shown. Enter `count=x` where x is between 1 and 99 to show the selected number of photo galleries.

CreatedBy
From version 6.2.0
	Used to filter by gallery creator. Enter
	`createdby=asdf`
	To display the galleries created by the journalist with username 'asdf'. Enter
	`createdby=#currentuser#`
	To display the user's own stories.

Days	By default, this option is not set. Enter `days=x` where x is between 1 and 99.
Excltaxonomywords **From version 6.2.0**	Exclude a specific taxonomy word. `Excltaxonomywords="11,12"`
IgnoreStatus **From version 6.2.0**	Use if the publish status should be ignored. The default is 0. `ignorestatus=1` Using it will display published and unpublished galleries.
Incltaxonomywords **From version 6.2.0**	Include a specific taxonomy word. `Incltaxonomywords="11,12"`
Objectclass	Sets the object class.

Order	Sets the sort order for the listing of photo galleries. By default the list is sorted by creation date. Enter
	`order=modified`
	to sort the list by the photo albums' modification date. Enter
	`order=published`
	to sort the list by the photo albums' publication date.

ShowExtraFields **From version 6.2.0**	Use to show extra fields. The default is 0. `showextrafields=1`
ShowTaxonomy **From version 6.2.0**	Use to show taxonomy words. The default is 0. `showtaxonomy=1`

PhotoGalleryLatest.pbo

This file is output when the system shows the latest galleries.

<%GalleryItems%>	This field outputs all items.

PhotoGalleryLatestItem.pbo

<%Category%>	The category for the photo gallery.
<%Counter%>	A counter for the gallery.
<%CreatedLongDate%>	The date on which the gallery was created, in long format.
<%CreatedShortDate%>	The date on which the gallery was created, in short format.
<%ImgLink%>	The img src for the first photo in the gallery.
<%Link%>	The link to the photo gallery.
<%ModifiedLongDate%>	The date on which the gallery was last modified, in long format.
<%ModifiedShortDate%>	The date on which the gallery was last modified, in short format.
<%PictureCount%>	The number of photos in the gallery.
<%Text%>	The description of the gallery.
<%ThumbnailLink%>	The link to the thumbnail version of the gallery.
<%Title%>	The title of the photo gallery.
<%UserSubmitDeleteLink%> **From version 6.2.0**	URL that if visited will delete the story - in case the story is a user submitted story and the user is the creator.
<%GalleryID%> **From version 6.2.0**	Contains the lopenr / StoryID of the gallery.
<%GalleryCategory%> **From version 6.2.0**	Same as Category.
<%GalleryDate%> **From version 6.2.0**	Contains the date of the gallery.
<%Location%> **From version 6.2.0**	Contains the location field value.

<%PhotoGalleryStatus%> **From version 6.2.0**	Contains the publish status of the photo gallery. It can be published (=1) or unpublished (=0). This is only returned if the photogallery-latest tag has the ignorestatus=1 parameter.
<%CreatedByUserID%> **From version 6.2.0**	Contains the creator's user ID if photo gallery is a blog entry.
<%CreatedByFirstName%> **From version 6.2.0**	Contains the creator's first name if photo gallery is a blog entry.
<%CreatedBySurName%> **From version 6.2.0**	Contains the creator's last name if photo gallery is a blog entry.
<%CreatedByEmailAdr%> **From version 6.2.0**	Contains the creator's email address if photo gallery is a blog entry.
<%BlogEntry%> **From version 6.2.0**	Whether the photo gallery is user generated (=1) or not (=0).
<%ISOCreatedDate%> **From version 6.2.0**	ISO formatted created date.
<%ISOModifiedDate%> **From version 6.2.0**	ISO formatted modified date.
<%Taxonomy%> **From version 6.2.0**	Contains the taxonomy words of the photo gallery. See PhotoGalleryLatestItemTaxonomy.pbo
<%ExtraFields%> **From version 6.2.0**	Contains the extra fields of the photo gallery. see PhotoGalleryLatestItemExtraFields.pbo
<%AvgRanking%> **From version 6.4.0**	Contains the average ranking of a photo.
<%RankingLink%> **From version 6.4.0**	The RankingLink does not contain the actual ranking value this must be added to the link by using client side scripting. The Following must be added to the link before submitting the link to the DLL '<%RankingLink%>' + '&rating=' + y Where y is the ranking given by a user.
<%TotalNoofVotes%> **From version 6.4.0**	Contains the total number of votes on the photo.

pbs photogallery-thumbnails

<pbs:photogallery-thumbnails>

This function uses the template **gen_tngallery.pbs**. This template is displayed by the action **/apps/pbcs.dll/tngallery**. Link to the gallery page from a story, or by using pbs photogallery-latest. If you want to link to the gallery manually, you must specify all parameters like this:

```
<a href="apps/pbcs.dll/tngallery?
*Date=20030609&Category=REPOSITORY&artno=609001&Ref=PH*
">
Link
</a>
```

Options

Columns	Determines the number of columns in the gallery table. By default, this is set to 3 columns. Enter `columns=x` where x is the number of columns in the gallery table.

ID	Specifies a gallery when you are viewing the gallery on another template, or through another action. This option has no default. Set this option as follows: `id="[<sitecode>,]<date>,<category>,<number>"`
Objectclass	Sets the object class used by the tag.

Rows	Determines the number of rows in the gallery table. By default, this is set to 3 rows. Enter
	`rows=x`
	where x is the number of rows. Setting
	`rows=0`
	will display all photos on the same page.

PhotoGalleryThumbnailItem.pbo

Output for each photo in the gallery.

<%Byline%>	The photo credit.
<%Caption%>	The caption for the photo.
<%Counter%>	The counter for the photo.
<%GalleryID%> **From version 6.4.0sr1**	The ID of the gallery (Lopenr).
<%GalleryItemLink%>	The link to the current photo in the browsable version of the gallery.
<%GalleryLink%>	The link to the first photo in the browsable version of the gallery.
<%GalleryZoomLink%>	A link to the current photo on a different template. The function is described in pbs link.
<%Height%>	The height of the photo.
<%ImgLink%>	The image src for the photo. It is used like this: `<img src="<%ImgLink%>&Maxw=200&Maxh=200">`
<%ItemNr%> **From version 6.4.0sr1**	The image number within the gallery.
<%RowEnd%>	This field has the value 1 on the last photo on the row. The field is used to end the table row. `[%{RowEnd}</tr>%]`
<%RowStart%>	This field has the value 1 on the first photo on the row. The field is used to start the table row. `[%{RowStart}<tr>%]`
<%TableEnd%>	This field has the value 1 on the last photo. The field is used to end the table. `[%{TableEnd}</table>%]`
<%TableStart%>	This field has the value 1 on the first photo. The field is used to start the table. `[%{TableStart}<table>%{code]`

<%ThumbnailGallerylink%>	A link to the thumbnail gallery. This field is used to display photos from the gallery on other templates, such as directly in a story.
<%\Title%>	The title of the photo.
<%Width%>	The width of the photo.

PhotoGalleryThumbnails.pbo

This file builds the wrapper for the gallery.

<%GalleryLink%>	A link to the browsable gallery. This field is used to display photos from the gallery on other templates, such as directly in a story.
<%GalleryTitle%>	The title for the album.
<%GalleryText%>	The description of the gallery.
<%Items%>	Outputs all the items.
<%NextLink%>	The link to the next page.
<%PageStart%>	The number for the first photo on the page.
<%PageStop%>	The number for the last photo on the page.
<%PrevLink%>	The link to the previous page.
<%ThumbnailGalleryLink%>	A link to the thumbnail gallery. This field is used to display photos from the gallery on other templates, such as directly in a story.
<%Total%>	The total number of photos in the gallery.

pbs ptshowimage

<pbs:ptshowimage>

This function is used to display one of the pictures in the gallery.

The tag might appear as follows:

```
<pbs:ptshowimage maximagewidth=468 showtitle=1 quality=100>
```

Templates

This option uses the template specified in the link to the page.

Options

Maximageheight	Sets the maximum height for the picture.
Maximagewidth	Controls the width of the picture. It must be present. If it is missing, the system will load the full image from the database, which could destroy your layout.
Quality	Controls the quality of the picture. The higher the number, the better the quality. The default quality setting is 85. This option can be set from 1 to 100, where 1 is the lowest quality and 100 is the best quality.
Showtitle	Determines whether the name of the picture will be overlayed on the picture. By default, the title will not be output. Enter `showtitle=1` to show the title.

Link from gallery

The link from the gallery is described in the section ImageItem.pbo.

Link from story

To link from the story to the zoomed picture, you must put the following code in the ArticlePicture.pbo:

```
<a href=
"<%pbcs.dll%>/misc?url=/misc/photos_single.pbs&IDato=<%ISODate%>&IKategori=<%Category%>&ID=<%ArtNo
>Zoom</a>
```

The URL must match the URL you have made for the zoom template. This function can only be used on the main picture.

ShowImage.pbo

This file is used by pbs ptshowimage and it can contain the following variables:

<%Byline%>	The photo credit.
<%Caption%>	The caption for the image.

Desktop background	To allow web guests to use the picture as a desktop background, you must use this piece of code to link to the functi
	`<a href="javascript:NewWindow(460,300,'/apps/pbcs.dll/misc?URL=/misc/photos_desktop.p`
	`&NoMal=1&IDato=<%IDato%>&IKategori=<%IKategori%>&ID=<%ID%>');">Set as desktop backgrc`
	You can change the name of the template used (bold in the example above), but not any of the other parameters. Th described in the section Set picture as desktop background.

<%ImagePath%>	The path to the image. It is used like this: `<img src="<%ImagePath%>" alt="" border="0">`
<%\Title%>	The title of the image.

Photo album export

This functionality enables export of one or more photo albums from the system. Output is static image files and an XML file describing the photo album structure and meta data to be delivered to an FTP server.

The exported photo album can be imported to other sites using the Album Importer.

Publicus.ini settings

To activate the permanent gallery, you must have the following setting in the Publicus.ini file in the GENERAL section section:

```
[General]
PhotoAlbumRootPath=\\imageserver\photoalbumexports
```

File structure

The export job will place the photo albums in a subfolder to the root path named

```
<site>-<date>-<category>-<PhotoalbumID>
```

where:

<category>	is the category of the photo gallery
<date>	is the date of the photo gallery
<PhotoalbumID>	is the ID of the album

<site>	is the site code of the site

If the <PhotoalbumID> folder already exists, it will be emptied before the exported files are copied to it.

The images are exported as JPG files and meta data is stored in an XML file named

```
<site>-<date>-<category>-<PhotoalbumID>-<PhotoId>
```

where:

<category>	is the category of the photo gallery
<date>	is the date of the photo gallery
<PhotoalbumID>	is the ID of the album
<PhotoId>	is the priority of the image.
<site>	is the site code of the site

The format of the XML file can be controlled by templates.

Object files

The following object files are used for photo album export:

- ExpPhotoAlbums.pbo (optional)
- ExpPhotoAlbum.pbo
- ExpPhotoAlbumItem.pbo

The export object files are part of the default edit templates; however, it is also possible to add user defined object files. These object files must be prefixed with a name (name_). For example:

- CCE_ExpPhotoAlbums.pbo (optional)
- CCE_ExpPhotoAlbum.pbo
- CCE_ExpPhotoAlbumItem.pbo

Using the mentioned syntax, it is possible to have multiple sets of templates for exporting photo albums. Upon export, users determine which templates to use through the **Template set** drop-down in the **Photo albums export** window. The path for user defined templates must exist in **web.dbo.siteconfig** table with the following values:

SITE	SECTION	NAME	VALUE
XX	SETTINGS	AlbumExportTemplates	\\templateserver\xx\AlbumExportTemp\

Where XX is the site ID and Value is the path to the user defined templates. The path is visible to the edit web server.

ExpPhotoAlbum.pbo

This is required. It includes the information for each photo album.

<%albumid%>	The photo album's unique ID.
<%AlbumPictures%>	A placeholder for each exported photo album item (image).
<%category%>	The category in which the photo album is placed.
<%createdate%>	The photo album's creation date.
<%createdby%>	The name of the user who created the photo album.
<%\date%>	The album's publication date.
<%lastmodifiedby%>	The name of the user who last modified the photo album.
<%lastmodifieddate%>	The date on which the photo album was last modified.
<%numberofimages%>	The total number of image in the photo album.

<%published%>	Indicates whether or not the album is published.
<%site%>	The ID of the Site on which the album is placed.
<%\summary%>	The photo album's summary.
<%\title%>	The photo album's title.

ExpPhotoAlbumItem.pbo

This is required. It includes the information about the photo album image. All images are exported in JPG format. The created image files are placed in the same directory as the export file.

<%credit%>	The name of the photographer or creator of the image.
<%imagecontentbase64%>	If enabled, the images are embedded in the XML file and the variable `*<%imagename%*` > will not have any value.
<% imageHeight%>	The height of the image.
<%imagemimetype%>	The image MIME type.
<%imagename%> (created file)	The name of the image file.
<%imageorder%>	The image's ranking in the photo album.
<%Imagewidth%>	The width of the image.
<%\summary%>	The image summary.
<%\title%>	The image title.

ExpPhotoAlbums.pbo

This is optional. If the **ExpPhotoAlbums.pbo** file exists, all the selected albums are exported to a single file. If the *ExpPhotoAlbums.xml *file does not exist, there will be one export file for each photo album.

<%Items%>	Outputs the selected photo albums.

Set picture as desktop background

This function allows the web guest to set an image from the database as his desktop background. You should include the following text to guide the user in this operation.

- Windows XP: Right-click on the photo to the right, and select *Set as desktop background*.
- Macintosh: Click and hold your mouse button down on the photo to the right. When the drop-down menu appears, select *Download Image to Disk*. From the Apple Menu, choose Control Panel > Appearance. Select the **Desktop** tab at the top of the window. Click *Place Picture* and choose the photo you downloaded. If you already have a picture set as your desktop, choose *Remove Picture* first and then click on *Place Picture*. Finally, select *Set Desktop.*

Link

To link to this function, use code from ShowImage.pbo.

Template

The template can be named whatever you like. You specify the template in the link, so be sure that you use the same name in both instances. The example uses **photos_desktop.pbs** in the /misc folder.

Tag

```
<pbs:ptshowimage maximagewidth=800 logo="/graphics/tbdotcom.gif" showtitle=1 size=1 quality=80>
```

Options

Logo	Sets which image should be overlayed on the picture as a logo.
Maximagewidth	Sets the maximum width of the image.
Quality	Sets the image quality.
Showtitle	Makes the system overlay the title on the picture. If no title exists, the image copyright will be used instead.
Size	Enter `size=1` to show a thumbnail of the picture, even if the full picture is loaded from the server.

Watermark images

The images in the system can be watermarked with either a text string or an image. This allows you to put a copyright text on your pictures, for example. You can reference the watermark in the URL of the **Pbcsi.dll** also known as the Image dll.

When using a separate image server, the copyright text must be present in the publicus.ini on the image server. If you are running a shared image server, you must use the option IMMAGGE

Watermark with object files

In the object file that is used to display the picture, you add the option that turns watermarking on.

Text watermark

This function allows you to output text on top of the picture.

Default	To add a text watermark to the image, add `&Title=1` after the image link like this: <img src="<%ImgLink%>&Title=1<%MaxWidth%>">. The system will use the text from the publicus.ini variable ImageCopyright as the watermark.
Option	Enter `title=2` to output the text with a black outline. This is done to make the text easier to read if the picture contains white areas. If the variable ImageCopyright is not set in the publicus.ini, the title of the picture will be used instead.

Image watermark

To add an image watermark to the image, add **&Logo=**... after the image link like this:

```
<img src="<%ImgLink%>&Logo=/folder/file.gif<%MaxWidth%>">
```

The system will place this GIF in the center of the image. Only GIF files can be used for a watermark.

The control the position of the watermark image, add **&LogoXPos=#** and **&LogoYPost=#** after the image link like this:

```
<img src="<%ImgLink%>&Logo=/folder/file.gif&LogoXPos=#&LogoYPos=#">
```

The X and Y positions are measured in pixels and use positive integers. By default, image watermarks are centered.

Watermark with tags

This can only be done on the gallery page where you use the tag pbs ptshowimage.

This function is deprecated. You should use the object file version instead.

Text watermark

To add a text watermark to the image, **Showtitle=1** inside the image tag like this: `<pbs:ptshowimage showtitle=1 quality=100>`. The system will use the text from the Publicus.ini variable GENERAL section#ImageCopyright as the watermark.

Image watermark

To add an image watermark to the image, add **Logo="/folder/file.gif"** after the image link like this: `<pbs:ptshowimage logo="/graphics/tbdotcom.gif">`. The system will place this GIF in the center of the image.

pbs ptimages

<pbs:ptimages>

The permanent gallery is used to build galleries that are available without any connections to a story. In the Online user interface, these images are administered through the **Misc** menu. Activating this functionality often requires a change to the Publicus.ini.

The gallery tag is also used to build image collections based on the day's published pictures. Please note that SAXOTECH Online also has a story-based photo gallery. This function is usually used when you have too many photos for a story. This is documented in the Article Tags chapter.

This tag requires object files.

Setup

To use the permanent gallery images, you must have Images=1 in the MODULES section section of the Publicus.ini. Please see the Publicus.ini chapter for more information about this setting.

Templates

This function can be used on all templates. See the example template permgall_ovr.pbs.

The tag must be inside table tags like this:

```
<table>
<pbs:ptimages>
</table>
```

Options

Columns	Determines how many columns the picture will span. You can set this option from 1 to 9.

Exclude	Allows you to exclude categories when you use the Today option to show the photos of the day. By default, this option is not used. Enter
	`exclude="NEWS%"`
	to exclude photos from the News category and all its subcategories. You can specify multiple categories to exclude in a comma separated list, like this:
	`exclude="NEWS,SPORTS,ART%"`
	You can use this option with the **Include** option.

Group	Determines which gallery the tag should show. The groups are managed through the editing interface. This option cannot be used with the Today option. This option has no default. To use this option, Specify the group by entering
	`group=GROUPNAME`
	Use
	`group=#CATEGORY#`
	to use the current category ID as group. Use
	`group=#MAINCATEGORY#`
	to use the main category ID for the current category as group.

Include	This option allows you to limit the categories when you use the Today option to show the photos of the day. By default, this option is not used. Enter
	`include="NEWS%"`
	to show photos from the News category and all its subcategories. You can specify multiple categories to include in a comma separated list, like this:
	`include="NEWS,SPORTS,ART%"`
	You can use this option with the **Exclude** option.

Maximageheight	Sets the picture height. There is no default setting. You can set this option by entering `Maximageheight=xxx` where xxx is any number between 10 and 9999 pixels.

Maximagewidth	Sets the picture width. There is no default setting. You can set this option by entering `Maximagewidth=xxx` where xxx is any number between 10 and 9999 pixels.
Order	Enter `order=modified` to sort the images with the most recently modified on top.
Quality	Controls picture compression. 1 is the lowest and 100 is the highest. By default, quality is set to 60.
Sortorder	By default the images are output with the oldest image first. By adding the sort option, `Order=modified` the images are sorted with the newest image first.
Today	Determines that the gallery should be built from today's pictures. This option cannot be used with the Group option. The date for which the gallery is shown can be controlled by adding `&Date=yyyymmdd` to the URL. This option will only output the main photo from a story. The photo must be in JPEG format.

ImageItem.pbo

<%articleparams%>	The necessary information to link back to the story from which the picture was taken. This function only works with the **Today** option.The field is used like this: `[%{ArticleParams}<a href="<%pbcs.dll%>/article?<%articleparams%>">Read story»</a>%]`
<%Byline%>	The photo credit.
<%Caption%>	The caption for the picture.
<%ImagePath%>	The path used in the `<img>` tag.
<%Title%>	The title of the picture.
<%StoryTitle%> **From 6.2.0**	The title of the story. Only available when using the "today" option.

Zoom link	The zoom link is built using multiple variables, like this: `<a href="/apps/pbcs.dll/misc?url=/misc/photos_single.pbs &Category=PERMGALL&IDato=<%IDato%>&IKategori=<%IKategori%>&ID=<%ID%>">` You can only change the category in this link.

ImageItemRowEnd.pbo

This file does not have any variables.

ImageItemRowStart.pbo

This file does not have any variables.

PhotoGalleryItemExtraFieldItem.pbo

From version 6.2.0

<%ExtraFieldName%>	Contains the name of the extra field.
<%ExtrafieldValue%>	Contains the value of the extra field.

PhotoGalleryItemExtraFields.pbo

From version 6.2.0

<%Items%>	Contains the extra field items.

PhotoGalleryLatestItemExtraFieldItem.pbo

From version 6.2.0

In addition, field specific object files are available using PhotoGalleryLatestItemExtraFieldItem<FieldName>.pbo

<%ExtraFieldName%>	Contains the name of the extra field.
<%ExtrafieldValue%>	Contains the value of the extra field.

PhotoGalleryLatestItemExtraFields.pbo

From version 6.2.0

<%Items%>	Contains the extra field items.

PhotoGalleryLatestItemTaxonomy.pbo

From version 6.2.0

<%Items%>	Contains the taxonomy words.

PhotoGalleryLatestTaxonomyItem.pbo

From version 6.2.0

<%TaxonomyID%>	Contains the taxonomy ID.
<%TaxonomyWord%>	Contains the taxonomy word.

PhotoGalleryTaxonomyItem.pbo

From version 6.2.0

<%TaxonomyID%>	Contains the taxonomy ID.
<%TaxonomyWord%>	Contains the taxonomy word.

Chapter 29 Newsletter

The Newsletter function allows you to send out a list of all the new stories for the day with links to the stories on your website.

The server must be set up to send out newsletters for your website. Otherwise, web guests can subscribe but no newsletter will be sent. The server setup can only be done by SAXOTECH.

- Newsletter template
  - Newsletter.txt
  - NewsletterAd.txt
  - NewsletterArticle.txt
  - NewsletterCategory.txt
  - NewsletterLinks.txt
  - NewsletterOAMSAdHTML.pbo
  - NewsletterOAMSAdSpot.txt
  - NewsletterSniff.txt
- pbs Newslad
- pbs newsletter_subscriptionform
  - NewsletterActivated.pbo
  - NewsLetterActivatedError.pbo
  - NewsletterRemoved.pbo
  - NewsLetterRemoveError.pbo
  - NewsLetterRemoveMail.pbo
  - NewsLetterSubscribeMail.pbo
  - NewsLetterSubscriptionForm.pbo
- Setup newsletters
- Subscription page with Sniff Search
- Web based management for newsletters
  - SubcatSimpleLink1.pbo

Newsletter template

Newsletter template files are used when sending the email to the subscriber. The HTML versions of the templates have HTML added before .txt, so Newsletter.txt becomes **NewsletterHTML.txt**.
All templates must be put in the **objects** folder.

It is important that you keep your lines under 60 characters in order for them to be handled correctly by email servers.

Newsletter.txt

The main file is called **newsletter.txt**. The file supports the following variables:

<%Ad1%>... <%Ad99%>	These fields will output the ads defined in the editorial system.
<%Ads%>	Outputs the ads for the newsletter.
<%Articles%>	Outputs the stories the subscriber has selected to receive.
<%Links%>	Outputs the links for the newsletter.
<%LongDate%>	Outputs the date the e-mail was generated.
<%UnsubscribeURL%>	Contains the link to the unsubscribe page.

NewsletterAd.txt

This file is output for every ad.

<%AdText%>	The ad text.
<%AdURL%>	The link to the ad.
<%HTMLAdText%>	The HTML version of the ad text.
<%\Title%>	The ad heading.

NewsletterArticle.txt

This file is called **NewsletterArticle.txt** and it is output for each story. It must be placed in the objects folder.

<%Category%>	The name of the category.
<%\Date%>	The story date in ISO format.
<%ImgLink%>	The link to the main photo attached to the story.
<%Leadin%>	The leadin for the story. To avoid the e-mail server destroying the email, you should add an extra space after each period. This example also shows how you can crop your leadin to a maximum of 300 characters. `<%Leadin$w(1,300)$r(".=. ")%>`
<%Link%>	The link to the story.
<%\Title%>	The title of the story.
<%SniffMatch%>	The Sniff word matched for the story. This field is only available on stories that have been output using Sniff words.

Special variables when using the keyword group file:

<%Category%>	The name of the category.
<%CategoryID%>	The category ID.
<%Super%>	The story kicker.
<%Text%>	The story's first text block.

Predefined **<newsl:ad>** variables:

If **nlsort** is not set, the variables **AdCategory** and **AdProfile** will be set corresponding to **NewsLetterHTMLCategory.txt**.

NewsletterCategory.txt

This file is called **NewsletterCategory.txt** and it is output for each category to which web guest has subscribed. It must be placed in the objects folder.

<%Articles%>	Outputs the stories in the category.
<%Category%>	Outputs the name of the category.

Predefined **<newsl:ad>** variables:

AdCategory	Corresponds to **<%Category%>** if the **nlsort** switch is not set.
AdProfile	If a profile corresponding to the category has been used to select stories it will be used in the tag. This means that the **nlsort** switch is not set, the *profiles *switch is set, the current category has an associated profile, and keywords are not used for the selection.

NewsletterLinks.txt

The file is output when the field **<%links%>** is used in the mail file. Links are added to the newsletter. The links are selected by a command line switch when the program that sends newsletters is run.

<%Description%>	The description of the link.
<%\Title%>	The title for the link.
<%URL%>	The URL for the link.

NewsletterOAMSAdHTML.pbo

Controls the way OAMS banners are placed in the composed list.

<%HTMLOAMSAd%>	The generated OAMS banner.

NewsletterOAMSAdSpot.txt

Controls the way an OAMS ad is called. The available variables are the same as for the **newsl:ad** tag described above.

<%AdLabel%>	The OAMS ad label, if applicable.
<%Categoryk%>	The category.
<%IDN%>	The spot identifier.
<%NoLink%>	Determines whether the link is included.
<%PageId%>	The Page ID.
<%Profile%>	The profile, if applicable.
<%Server%>	A reference to the OAMS server.
<%Site%>	The site.
<%Spot%>	The spot.
<%SubLabel%>	The sublabel, if applicable.
<%Subsite%>	The subsite.

NewsletterSniff.txt

<%IsFirstItem%>	Is 1 when processing the first sniff item.
<%Leadin%>	The leadin for the story. To avoid the e-mail server destroying the e-mail, you should add an extra space after each period. This example also shows how you can crop your leadin to a maximum of 300 characters. `<%Leadin$w(1,300)$r(".=. ")%>`
<%Link%>	The link to the story.
<%SniffMatch%>	The Sniff word matched for the story.
<%\Title%>	The title of the story.

Predefined **<newsl:ad>** variables:

If **nlsort** is not set, the variables **AdCategory** and **AdProfile** will be set corresponding to **NewsLetterHTMLCategory.txt**.

pbs Newslad

<newsl:ad>

It is possible to put OAMS ads into newsletters using **newsl:ad** tags in the HTML versions of the templates. For many of the variables, default values can be defined in the **newsletter.ini** in the **ADX** section. In some cases, some of the variables are predefined. Specific instances are described for each template. The predefined values can be overridden in the tag definition.

AdCategory	The category to use on the spot. Ini file ADX variable: **DefaultCategory**. Default value: `-` but the value should be defined.
AdLabel	Instead of using category or profile use a direct OAMS label. Ini file ADX variable: **DefaultLabel**. Default value: `-` (no label).
AdPageID	An ID common to all OAMS references in the newsletter. Must be unique on the OAMS server. Default: A random ID is generated.
AdProfile	The profile to use on the spot. Ini file ADX variable: **DefaultProfile**. Default value: `-1` (no profile).
AdServer	Name of the server that should handle OAMS calls. Ini file ADX variable: **Server**.
AdSite	The site code used in the OAMS calls.Ini file ADX variable: **Site**. Default: The site defined by the `-v` switch.
AdSubLabel	The ad sublabel. Ini file ADX variable: **DefaultSubLabel** Default: No sublabel.
AdSubSite	The subsite used in the OAMS calls. Ini file ADX variable: **SubSite**. Default value: `001`
IDN	An ID identifying the individual hole. Must be unique on the OAMS server. Default: A random ID is generated.
NoLink	If this is 1, no link will be generated for the OAMS ad and only the banner will be shown. Ini file ADX variable: **NoOamsLinks**. Default value: `0`
Spot	The spot to replace the newsl:ad tag. Ini file ADX variable: **DefaultSpot** Default value: `-` but the value should be defined.

pbs newsletter_subscriptionform

<pbs:newsletter_subscriptionform>

The email based management is a simpler version of the subscription form. This version allows the web guest to enter an email address without registering. An email is sent to the web guest when he subscribes or unsubscribes. This email contains a link he must click to confirm his subscription status. When this option is used for managing newsletters, the web guest cannot change the categories to which he subscribes.

Setup

NewsletterMailID in Publicus.ini sets the sender name for the confirmation emails.

Template

This function can be used on all templates. See the example file Newsletter_ovr.pbs.

NewsletterActivated.pbo

Used when the web guest subscribes to the newsletter successfully. This file has no fields.

NewsLetterActivatedError.pbo

Shown if there is an error when the web guest tries to subscribe to the newsletter. This file contains no fields.

NewsletterRemoved.pbo

Used when the web guest unsubscribes successfully. This file has no fields.

NewsLetterRemoveError.pbo

Shown if there is an error when the web guest tries to unsubscribe to the newsletter. This file contains no fields.

NewsLetterRemoveMail.pbo

Used to send the confirmation email to a web guest who is unsubscribing.

<%url%>	The URL the user must click to confirm the unsubscription.

NewsLetterSubscribeMail.pbo

Used to send the confirmation e-mail to the subscriber.

<%url%>	Contains the URL the user must click to confirm the subscription.

NewsLetterSubscriptionForm.pbo

The NewsletterSubscriptionForm.pbo object file describes the form that users use to subscribe to newsletters.

An example subscription form frame

```
<script>
  function SaveEmail() {
    newsletterform.Remove.value="0";
    newsletterform.Subject.value="Subscribe to the newsletter";
    newsletterform.submit();
  }

  function RemoveEmail() {
    newsletterform.Remove.value="1";
    newsletterform.Subject.value="Unsubscribe to the newsletter";
    newsletterform.submit();
  }
</script>

<form name="newsletterform" action="<%pbcs.dll%>/savenewsletteruser" method="POST">
  Form fields goes here.
</form>

<a href="javascript:SaveEmail();">Subscribe</a><br>
<a href="javascript:RemoveEmail();">Unsubscribe</a><br>
```

Available form fields

Add	This hidden field can be used for telling if the categories in the **categories** field are to be added (=1) to the list of subscribed categories for the email address or if they are to replace (=0) the list. Default is replace (=0). `<input type="hidden" name="Add" value="1">`
Categories	This hidden field is used to set the categories to which the web guest subscribes. `<input type="hidden" name="Categories" value="NEWS01,NEWS02">`
Email	Captures the subscriber's e-mail address. `<input name="Email" size="30" />`

HTML	These radio buttons enable the subscriber to choose between HTML and plain text newsletters.
	
	`<input type="radio" name="HTML" value="0" checked>`
	
	Plain text
	
	`<input type="radio" name="HTML" value="1">`
	
	HTML mail

RedirectURL	This hidden field is used to set the page to which the subscriber will be directed after the form is processed. `<input type="hidden" name="RedirectUrl" value="<%pbcs.dll%>/misc?url=/mal/NewsLetterMailSent.pbs">`
Remove	This hidden field is used to control whether the web guest is subscribing or unsubscribing. This field is filled in by the JavaScript. `<input type="hidden" name="Remove">`
Subject	This hidden field is used to set the subject of the e-mail. This field is filled in by the JavaScript. `<input type="hidden" name="Subject">`

Setup newsletters

Newsletter.ini file

The file that controls the newsletter is called newsletter.ini. This file must be placed in the objects folder. The file has the following sections:

SETTINGS	This is the header for the Settings section.
	• CharSet - Sets the character set of the email. It is set like this:
	`CharSet=iso-8859-1`
	• Language - Sets the correct language for use in the date, for example. The option is used like this:
	`Language=DEN (Danish)`
	`Language=SWE (Swedish)`
	`Language=GER (German)`
	• ReplyTo - Sets the reply to address for the e-mail. The option is used like this: `ReplyTo=sendereditor@tahoe.com`* Sender - Sets which email address the newsletter is sent from. The option is used like this:
	`Sender=info@domain.com`
	• URL - The URL is used to set the base URL for your site. The option is used like this:
	`URL=http://www.nevadaappeal.com`

SYSTEM	This is the header for System section. It must be present.
	• Antall - Sets the maximum number of stories sent from a single category.
	• DateFormat - Sets the date format. This field uses the standard SAXOTECH Online date formatting options. The option is used like this:
	`DateFormat=dddd, mmmm dd, yyyy`
	• From - Sets the sender of the e-mail. The option is used like this:
	`From=nevedaappeal.com`
	• Sniff categories - By default the Sniff Search looks in all categories for the system. To limit Sniff Search categories, use this option like this:
	`SniffCategories=NEWS,SPORT,...`
	• Subject - Sets the subject of the e-mail. The option is used like this:
	`Subject=Your email newsletter from nevadaappeal.com`

Subscription page with Sniff Search

This subscription page is a more advanced version, which allows the web guest to select stories based on keywords. This is an extension of the basic subscription feature.

Sniff Search

Sniff search is a function that allows a web guest to receive links to all stories with a specific keyword. This functionality can allow Bob's Widget Company to receive all stories in which widgets are mentioned.

Types

The added search word can be one of three types.

Containing	A word in the story must contain the word. These words are marked with [] in the list of words. If the sniff word is tree, it would match *tree, treehugger* and *street*.

Matching	The word must match a word in the story exactly. If the sniff word is *tree*, it would match tree, but not *treehugger* or *street* .
Starting with	A word in the story must begin with the selected word. These words are marked with <> in the list of words. If the sniff word is *tree*, it would match *treehugger* and *tree*.

Template

This function also uses the _**subscribe.pbs** template. See the example template Sniff_Subscribe.pbs.

Fields

JavaScript	This version of the form requires that all the JavaScripts in the example are present. You should cut and paste the example form and work from there.
Sniff	Used to store the selected sniff words. `<input type="hidden" name="sniff" value="<pbs:#sniff#>">`

Web based management for newsletters

Web-based management of newsletters lets web guests manage their subscriptions through a webpage. When using web-based management the web guest can choose categories from which to receive stories.

User registration

The web guest must be a registered user of your site. This is because the newsletter uses the same database as the White Pages for user registration.

User registration is described in the section Chapter 25 User Registration.

Setup

The setup is done using the SAXOTECH Online user interface. For more information, see the *SAXOTECH Online User Manual.*

You should create a category called NLETTER in which you set **Link type** to **subscribe** and **registration** to either **Simple** or **Advanced**. This affects which registration template the web guest will see.

Link to the page

You can link to the page using the ft tag. It will use the link type from the category to create the correct link. If you are going to link to the page manually, use the following link:

```
/apps/pbcs.dll/subscribe?Category=NLETTER
```

Template

The template is called **gen_subscribe.pbs**. You must create a template specifically for the category. If the category you have created is called NLETTER, call the template **nletter_subscribe.pbs**. If you call the template gen_subscribe, web guests can bypass the user registration screen by putting another category name in the URL.

Tags and fields

The following tags and fields are available on the subscription page. Please see the example file **nletter_subscribe.pbs** in the sample template zip file.

Active	Determines whether the subscription function is active. `<input type="hidden" name="Active" value="<pbs:#active#>">`

Category subscribed	This tag outputs the value of the category to which the web guest has subscribed. `<input type="checkbox" name="Category" value= "BUSINESS" <pbs:#CB_BUSINESS#>> Business `
First name	Outputs the subscriber's first name. `<pbs:#fname#>`
HTML format	Outputs the value checked if the subscriber has selected to receive HTML mail. `<input type="checkbox" name="HTML" value=1 <pbs:#ReceiveHTML#>>`
Received letters	This tag shows how many newsletter the subscriber has received. `<pbs:#ReceivedLetters#>`
Surname	Outputs the subscriber's surname. `<pbs:#sname#>`

Basic subscription page

Form code	The form must begin with:
	`<form name="selections" action="<pbs:prog>/subscribe?UserReg=1" method= "POST">`
	and must end with:
	`</form>`

JavaScript	The JavaScript present in the example code must be on the page.
Listing categories	You must put checkboxes for the available categories in the form. Each checkbox must have the name Category, and the value must be the category ID for the desired category. `<input type="checkbox" name="Category" value="NEWS" <pbs:#CB_NEWS#>>News  <input type="checkbox" name="Category" value="SPORTS" <pbs:#CB_SPORTS#>>Sports  <input type="checkbox" name="Category" value="OPINION" <pbs:#CB_OPINION#>>Opinion ` The pbs tag outputs the checkbox value when a subscription is edited. This tag must be present for each category.

Select/unselect all	The following code is needed if you want a link to subscribe/unsubscribe to all subcategories of a category: `<a href="javascript:SelectAll(true, 'NEWS');">Select all</a> -` `<a href="javascript:SelectAll(false, 'NEWS');">Unselect all</a> ` You must replace NEWS with the category ID of the category to which the links should apply.

Subscribe/unsubscribe	The following piece of code is needed to output the links that allows the user to subscribe or unsubscribe:
	
	`<a href="javascript:doSubmit(0);">Subscribe</a> -`
	`<a href="javascript:doSubmit(1);">Unsubscribe</a>`

Unsubscribe field	The form must contain: `<input type="hidden" value="0" name="unsubscribe">` Without this code, the form will not work.

Listing Subcategories

The following code can be used to output the checkboxes for the categories. This code is used to automatically output the checkboxes for all subcategories. This can also be done manually:

```
<pbs:subcategories maincategory="NEWS" class=1 force exclude="NEWS13">
```

This code uses the subcategory function.

Class	The class that determines which object file will be used.
Exclude	You can exclude categories from the listing by entering their category IDs here.
Force	Forces the listing of all subcategories. It is needed to output the subcategories without stories on today's date.
Maincategory	The main category for which you want to list subcategories.

SubcatSimpleLink1.pbo

The subcategory in the example code uses SubcatSimpleLink1.pbo. It must contain the following codes. Please see the example file on how to use these variables.

<%Categoryid%>	This field must be placed inside the value field of the input checkbox.
<%Categoryname%>	Outputs the name of the category.
<%Checked%>	This variable will mark the checkbox if the web guest has subscribed to the category.

Example

```
<input type="checkbox" name="Category" value="<%CategoryID%>" <%Checked%>>
<%CategoryName%><br>
```

Chapter 30 Newsletter Composer

The Newsletter Composer function allows you to send newsletters for which you select the stories. This newsletter can be targeted based on demographic information and on subscription categories.

Setup

The server must be set up to send out newsletters for your website. If this is not done, you can create the newsletters, but they will not be sent. Your server administrator must set up the routine that actually sends the e-mails out to the subscribers.

Newsletter.ini

The file that controls the newsletter is called **newsletter.ini**. This file must be placed in the objects folder. For more information, see the section newsletter_Newsletter#Newsletter.iniFile.

Templates

These files are used when sending the e-mail to the subscriber. The HTML versions of the templates have HTML added before .pbo, so NewsletterArticle.pbo becomes NewsLetterArticleHTML.pbo. All templates must be put in the objects folder.

> ⚠ It is important that you keep your lines under 60 characters in order for them to be handled correctly by the email servers.

- Main template
- NewsletterAd.pbo
- NewsletterArticle.pbo
- Newsletter email
- Web based management

Main template

The main template is coded directly in the newsletter composer. The following fields are available.

<%LongDate%>	Outputs the date field.
<%NewsletterContent%>	Outputs all ads and stories.
<%UnsubscribeURL%>	Contains an URL used to unsubscribe. The unsubscribe link is only valid for newsletters sent based on categories. It should be put in a conditional like this: `[%{SubscriptionType=0}` `<a href="<%UnsubscribeURL%>">` `Unsubscribe` `</a> ` `%]`

NewsletterAd.pbo

This file is output for every ad.

<%AdText%>	The ad text.
<%AdURL%>	This field contains the value of the URL field on the ad.
<%\Title%>	The ad heading.

NewsletterArticle.pbo

Output for each story. It must be placed in the objects folder.

<%Category%>	The name of the category.
<%\Date%>	The publishing date for the story.
<%ImgLink%>	The link to the main photo attached to the story.

<%Leadin%>	The leadin for the story. To avoid problems with the e-mail server, you should add an extra space after each period. This example also shows how you can crop your leadin to a maximum of 300 characters. `<%Leadin$w(1,300)$r(".=. ")%>`
<%Link%>	The link to the story.
<%Lopenr%>	The story's ID number.
<%\Title%>	The title of the story.

Newsletter email

Getcontent

Retrieves content from another website. You can have more than one `getcontent` tag. The tag can be used with newsletter composer in both the plain text and HTML version of the newsletter. The tag is defined as follows:

```
<newsl:getcontent url="http://www.helloworld.com/apps/pbcs.dll/section?category=newsl>
```

or

```
<newsl:getcontent removehtmlheader=1 url="http:
//www.helloworld.com/apps/pbcs.dll/section?category=newsl>
```

removehtmlheader	Determines whether HTML header information should be removed. It is also possible to activate the remove for all `getcontent` tags by calling **newsletter.exe** with the parameter `-removehtmlhead`
URL	The URL from which content will be retrieved. If the URL is not valid, the tag will be empty and the newsletter will not be sent.

To activate `getcontent` in the newsletter, the **newsletter.exe** must be called with the parameter

```
-allowgetcontent
```

If the tag fails, the newsletter will not be sent. The application will try to resend the newsletter during the next run.

Web based management

Web-based management of newsletters lets subscribers manage their subscriptions through a webpage. This can be done on your standard newsletter subscription page. You should name your category something that will make it easy to see that is a newsletter composer category. The value you specify will be shown in the newsletter composer interface

Example code

This code goes on your subscription page.

```
<input type="checkbox" name="Category" value="COMPFOOTBALL" <pbs:#CB_COMPFOOTBALL#>> Football
roundup<br>
<input type="checkbox" name="Category" value="COMPARTS" <pbs:#CB_COMPARTS#>> Arts newsletter<br>
```

User registration

The web guest must be a registered user of your site. This is because the newsletter uses the same database as the White Pages for user registration. User registration is described in the section Chapter 25 User Registration.

Chapter 31 Omniture

SAXOTECH Online web site statistics information is provided through a web analytics engine called SiteCatalyst powered by Omniture.

- pbs omniturecode
  - URL action
- Setup Omniture

pbs omniturecode

<pbs:omniturecode>

In order for Omniture to gather website statistics, it must find Omniture JavaScript on the pages it should analyse. If you would like to determine on which pages and where on the pages to insert Omniture JavaScript, you should use the Omniture tag pbs omniturecode.

Checks will be made to ensure that the code will only be inserted once per page at the bottom of the page.

URL action

In order to exclude Omniture JavaScript from the response of certain actions, you can add `&omniture=0` to the URL. This will work for all responses of actions into which the Omniture JavaScript is normally inserted.

In order to exclude Omniture JavaScript from the response of the story and section actions, you can add `&mime=xml` to the URL. For example, this can be used if NITF Stories or RSS feeds are requested.

Setup Omniture

To use this tag, the following settings must be made in the Publicus.ini file in the OMNITURE section:

```
[OMNITURE]
UseOmnitureTag=1
SuiteID=saxotechcom
BuildHierarchy=0
CookieDomainPeriods=2
```

The `UseOmnitureTag=1` setting disables the automatic insertion of the Omniture JavaScript. Instead, it inserts Omniture JavaScript wherever an Omniture tag is found in the templates.

Please note that the following actions (on other tags) are not affected by this setting: Tip a Friend, AdClick, PollClick and NewsLetterSignup. They will generate statistical information anyway.

Because Omniture SiteCatalyst will only generate statistics if there is Omniture JavaScript on the pages, we recommend not using this feature except in special circumstances.

Chapter 32 US Weather

To be able to show weather on your website, you need to have a feed from a weather provider. SAXOTECH Online only offers the tools to display the weather information on your site. We do not provide weather information.

The system does not know which measurement system is being used. The system does not see the difference between Fahrenheit and Celsius, so you must use the same measurement system as your weather provider.

There are multiple sets of weather tags. The weather tags starting with "Accu" are only used in the US.

- pbs accuwforecast
  - AccuWForecast.pbo
- pbs accuwcurrent
  - AccuWCurrent.pbo
- pbs accuwname
- pbs accuwmap
  - AccuWMap.pbo
- pbs accuwoptions

pbs accuwforecast

<pbs:accuwforecast>

The forecast function displays the forecasted weather.

Options

Label	Used to select the geographical location code.
Dateformat	Sets the date format. This option follows the default SAXOTECH Online behavior and can be set like this: `dateformat="h:mmam/pm"`
Class	Sets the object class for the tag.

Part	Determines which record in the forecast table is used. Enter
	`part=0`
	to show the first record. Enter
	`part=1`
	to display the second record. The information displayed depends on your weather provider. Enter
	`part=-1`
	to display all records.

Local	Determines which table will be used in the database. If the weather provider delivers more details for your local region, you can use this table instead by entering `local=1` If this option is not set, the forecast will use the default national table.
World	Used to select data from outside the US. The option is used like this: `world=1`

AccuWForecast.pbo

<%Updated%>	The update time for the weather.
<%CurrLabel%>	The forecast label.
<%Name%>	The forecast name.
<%Part%>	The part ID for the forecast.
<%Icon%>	The icon number for the weather. It is usually used like this: `<img src="/graphics/weather/icons/<%Icon%>.gif">`

<%ShortDesc%>	The short description of the weather. This is usually a single word like sunny, foggy or overcast.
<%DayDesc%>	The description of the day.
<%High%>	The projected high temperature in Fahrenheit.
<%HighCelsius%>	The projected high temperature in Celsius.
<%Low%>	The projected low temperature in Fahrenheit.
<%LowCelsius%>	The projected low temperature in Celsius.
<%WindSpeed%>	The wind speed.
<%WindDir%>	The wind direction.
<%WindGusts%>	The projected wind gusts.
<%Humidity%>	The humidity value.
<%Dewpoint%>	The dew point.
<%Barometer%>	The barometric pressure.
<%Visibility%>	The visibility information.
<%State%>	The state code for the entry being viewed.
<%MapCode%>	The map code for the entry being viewed.

pbs accuwcurrent

<pbs:accuwcurrent>

This function gives you the current weather.

The tag must be used like this:

```
<pbs:\!accuwcurrent label="TOL" class=1>
```

Options

Label	This option is used to select the geographical location code. The code must be in the system's option table, and Accuweather must transfer information for the code to the server. You can also set the label in the URL by adding `&Label=XXX` to the URL.
Locklabel	Used to lock the label for the tag so it cannot be overridden from the URL. This is useful when you have a weather symbol in the header that should not change. By default this option is not used. To use locklabel enter `locklabel=xxx` in the tag instead of Label.
Class	Sets the object class for the object file.
World	Used to select data from outside the US. The option is used like this: `World=1`

StateInfo	Makes the tag list all cities in the selected state using the Label option. By default this option is not used. To use StateInfo enter `Stateinfo=1`
Selection	Used to specify multiple city codes. Multiple codes must be specified using a semicolon separated list.

AccuWCurrent.pbo

Not all vendors of weather content provide all the fields listed.

<%Updated%>	The update time for the weather.
<%CurrLabel%>	The current label.
<%Name%>	The current name.
<%ShortDesc%>	A short description of the weather. This is usually a single word like sunny, foggy or overcast.
<%Icon%>	The icon number for the weather. It is usually used like this: `<img src="/graphics/weather/icons/<%Icon%>.gif">`
<%CurrTemp%>	The current temperature.
<%FeelsLike%>	The feels-like temperature.
<%Humidity%>	The humidity value.
<%Dewpoint%>	The dew point.
<%Barometer%>	The barometric pressure.
<%WindSpeed%>	The wind speed.
<%WindDir%>	The wind direction.
<%WindCurr%>	The current wind speed.
<%WindGusts%>	The current wind gusts
<%Visibility%>	The visibility information.
<%Sunrise%>	The sunrise time.
<%Sunset%>	The sunset time.
<%State%>	The state code for the entry being viewed.
<%MapCode%>	The map code for the entry being viewed.

pbs accuwname

<pbs:accuwname>

This function outputs the name associated with a label.

Options

Label	The name of the location for which you would like to show weather. This is the same label you use to show current conditions and forecasts.
Country	Sets the country code for the label. By default the country code is The United States. Other country codes can be used. **Example:** C019 = Bermuda

pbs accuwmap

<pbs:!accuwmap>

This function allows you to display maps provided by the weather service.

The{{<pbs:!accuwmap>}} tag can be placed on all templates.

Options

Maplabel	Selects the map from the database. Which maps are available depends on the weather service you use. Please see the list provided by your weather provider. The following maps are provided by AccuWeather: • LOCAL_RAD - Local radar. • REGRAD - Regional radar. • USRAD - US Radar. • TODHI - Today's high temperatures. • TONLO - Tonight's low temperatures. • REGSAT - Regional satellite. • USSAT - US Satellite.
Dateformat	Sets the date format. This option follows the default SAXOTECH Online behavior and can be set like this: `dateformat="h:mmam/pm"`
Maxwidth	Sets the maximum width of the map in pixels.

AccuWMap.pbo

<%Map%>	Contains the path to the map. It is use in an image tag like this: `<img src="<%Map%>" alt="" border="0">`

pbs accuwoptions

<pbs:accuwoptions>

This tag can be used to build a search form that allows web guests to browse states and cities.

The {{<pbs:accuwoptions> }}tag can be used on all pages. The tag must be used uncached.

Options

States	This option makes the tag list states.
Cities	This option makes the tag list cities.
States	This option makes the tag list states.
States	This option makes the tag list states.
Cities	This option makes the tag list cities.

Example search form

```
<form name="weathersearch" action="<pbs:prog>/section?Category=WEATHER2" method="POST">
<input type="hidden" name="LastState" value="<pbs:!#State#>">
<input type="hidden" name="MapLabel" value="<pbs:!#MapLabel#>">
<table border=0>
<tr>
<td><b>Choose state: </b></td>
<td colspan=2>
<select name="State" OnChange="javascript:document.weathersearch.submit();">
<pbs:!accuwoptions type=states>
</select>
</font>
</td>
</tr>
<tr>
<td><b>Choose city: </b></font></td>
<td colspan=2>
<select name="Label" OnChange="javascript:document.weathersearch.submit();">
<pbs:!accuwoptions type=cities>
</select>
</td>
</tr>
</table>
</form>
```

Chapter 33 European Weather

These tags are used to present weather information provided by Storm Weather Services, Cirrus, DNMI and DDMI.

Templates

The European Weather tags can be used on all templates.

- pbs weatherforecast
  - WeatherForecast.pbo
  - WeatherForecastColItem.pbo
  - WeatherForecastItem.pbo
  - WeatherForecastRowItem.pbo
- pbs weathergraph
  - WeatherGraph.pbo
- pbs weatherlastupdate
- pbs weatherobservations
  - WeatherObservation.pbo
  - WeatherObservationItem.pbo
- pbs weatherpicture
  - WeatherPicture.pbo
- pbs weathertext
- pbs weathertidesunmoon
  - WeatherTideSunMoon.pbo
- Storm weather service

pbs weatherforecast

<pbs:weatherforecast>

This tag can be used on all templates.

The <pbs:weatherforecast> tag can be used in the following manner:

```
<pbs:weatherforecast days=4 location=64 site=TA>
```

Options

Days	Determines for how many days into the future the tag should show forecasts.

Location	Used to select the location ID for the information.
Site	This option is used to set the site code from which the weather data will be retrieved. This option is only to be used if you are retrieving information from another site.

WeatherForecast.pbo

<%WeatherForecastContent%>	Lists all the content.

WeatherForecastColItem.pbo

If no region information is used, the weather forecast uses these column and row tags.

<%WeekDay%>	Shows the day of the week.

WeatherForecastItem.pbo

<%WeekDay%>	The day of the week.
<%WCode%>	
<%MaxTemp%>	The maximum temperature.

WeatherForecastRowItem.pbo

If region information is used, the weather forecast uses this tags.

<%Name%>	The region name.
<%ColContent%>	Shows the content of the column.

pbs weathergraph

<pbs:weathergraph>

Averages are stored on the year 9999 in the table LocationForecast.

This tag can be used on all templates.

Options

Objectclass	Sets the object class for the tag.
Site	Sets the site code from which the weather data will be retrieved. This option is only to be used if you are retrieving information from another site.
Location	Used to select the location ID for the information.
Dateformat	Sets the date format for the graph. It uses default SAXOTECH Online formats. By default, the format is "d.m."
Dateoffset	Sets the offset from today's date.
Days	Sets how many days the graph should show. The default is 30 days.
Type	Sets the type of graph.
Precip	This option sets the graph to show precipitation.
Temp	With this setting the graph shows temperature. This is the default.

WeatherGraph.pbo

<%ImgLink%>	Contains the link to the graph image.

pbs weatherlastupdate

<pbs:weatherlastupdate>

This tag is used to show the latest time that weather information was updated. This tag can be used on all templates.

Options

Format	Sets the date format. This option uses the standard SAXOTECH Online method of formatting dates.
Region	Used to select the region for which you would like to show the update. Which region codes match which areas is dependant on the server configuration. Contact the server administrator for a list of relevant region codes.
Type	By default, this is not used and the tag shows the update time from either WeatherRegionForecast or WeatherLocationForecast, depending on which was updated most recently. Set `type=observations` to force the tag to show either WeatherRegionForecast or WeatherLocationForecast.

pbs weatherobservations

<pbs:weatherobservations>

This tag is used to report weather observations. This tag can be used on all templates.

Options

Region	Used to select the region for which you would like to show observations. Which region codes match which areas is dependant on the server configuration. Contact the server administrator for a list of relevant region codes.
Site	Sets the site code from which the weather data will be retrieved. This option is only to be used if you are retrieving information from another site.

WeatherObservation.pbo

<%RegionName%>	The region name.
<%ObservationItems%>	Lists all the items.

WeatherObservationItem.pbo

<%Name%>	The region name.
<%WCode%>	
<%MaxTemp%>	The maximum temperature.
<%WdDir%>	The wind direction.
<%WdSpeed%>	The wind speed.

pbs weatherpicture

<pbs:weatherpicture>

This tag is used to display a weather image. This image can be a map, satellite image, or other image provided by the weather service. This tag can be used on all templates.

Options

Region	Used to select the region for which you would like to show the picture. Which region codes match which areas is dependant on the server configuration. Contact the server administrator for a list of relevant region codes.
Width	Sets the width of the weather picture.
OnclickUrl	Used to open up a larger version of the image. It is used like this: `<pbs:weatherpicture region=8 width=235 onclickurl= "javascript:NewWindow(536,434,'<%imgurl%>&maxw=500')">`

WeatherPicture.pbo

<%Picture%>	This field contains the picture URL.

Example

```
<%Picture%><br>
```

pbs weathertext

<pbs:weathertext>

This tag is used to retrieve the weather forecast text and the description of the current weather. This tag can be used on all templates.

Options

Region	Used to select the region for which you would like to show the forecast. Which region codes match which areas is dependant on the server configuration. Contact the server administrator for a list of relevant region codes.
Dateoffset	Determines for how many days into the future you would like to show the forecast.
Site	Sets the site code from which the weather data should be retrieved. This option is only to be used if you are retrieving information from another site.

pbs weathertidesunmoon

<pbs:weathertidesunmoon>

This tag can be used on all templates.

WeatherTideSunMoon.pbo

<%TideHigh%>	The time when the tide is highest.
<%TideLow%>	The time when the tide is lowest.
<%Sunrise%>	The time the sun rises.
<%Sunset%>	The time the sun sets.
<%Moonrise%>	The time the moon rises.
<%Moonset%>	The time the moon sets.
<%MoonPhaseType%>	The phase of the moon.
<%MoonPhaseTime%>	The time of the moon phase.

Storm weather service

Storm weather service provides a different type of service and has its own separate tags.

Object files from detskjer.no:

```
<pbs:weatherpicture region=8 width=235 onclickurl=
"javascript:NewWindow(536,434,'<%imgurl%>&maxw=500')">
<pbs:weathertext region=3 site=TA><br>
<pbs:weatherlastupdate format="dd. mmmm yyyy kl. hh:nn">
<pbs:weatherobservations site=TA region=1>
<pbs:weatherlastupdate format="dd. mmmm yyyy kl. hh:nn" region=8 type=observations>
<pbs:weathertext region=8 dateoffset=3><br>
<pbs:weatherforecast days=4 location=64 site=TA>
```

Chapter 34 Horoscopes

The horoscope function is used to display horoscope data imported from a horoscope provider.

- pbs horoscope

pbs horoscope

<pbs:horoscope>

Template

The tag can be put on all templates. The example uses the template **horoscope_ovr.pbs**.

Options

Category	The category option is used to select the data for the different signs: • Today - This option makes the tag show the extended horoscope for today. • Aquarius - This option makes the tag show the horoscope for Aquarius. • Pisces - This option makes the tag show the horoscope for Pisces. • Aries - This option makes the tag show the horoscope for Aries. • Taurus - This option makes the tag show the horoscope for Taurus. • Gemini - This option makes the tag show the horoscope for Gemini. • Cancer - This option makes the tag show the horoscope for Cancer. • Leo - This option makes the tag show the horoscope for Leo. • Virgo - This option makes the tag show the horoscope for Virgo. • Libra - This option makes the tag show the horoscope for Libra. • Scorpio - This option makes the tag show the horoscope for Scorpio. • Sagittarius - This option makes the tag show the horoscope for Sagittarius. • Capricorn - This option makes the tag show the horoscope for Capricorn.

Chapter 35 TV Guide

The TV Guide Component is used to display the TV guide. To use this component you need a feed from an information provider.

You should use a style sheet when you build the TV guide. If not, your pages will be very large. The examples in this documentation use style sheets.

The style sheet used by the TV guide is called tv.css and is located in the folder* /includes.*

- pbs tv_getoptions
- pbs tv_LineUpData
  - TVLineUpSelect.pbo
  - TVLineUpZipSelect.pbo
- pbs tv_programinfo
  - TvPrograminfo.pbo
- pbs tv_schedule
  - TVSchedule.pbo
  - TVScheduleHeaderRow.pbo
  - TVScheduleRow.pbo
  - TVScheduleRowCell.pbo

pbs tv_getoptions

<pbs:tv_getoptions>

This tag extracts information from the database.

Template

The search form can be put on any template. The results page must be set to the page used for showing the line-up.

The example is placed on the template **tv_ovr.pbs**.

These hidden fields are used to store information when the web guest browses the TV Guide:

```
<input type="hidden" name="Provider" value="<pbs:!#Provider#>">
<input type="hidden" name="LineUp" value="<pbs:!#LineUp#>">
<input type="hidden" name="Tier" value="<pbs:!#Tier#>">
<input type="hidden" name="Interval" value="<pbs:!#Interval#>">
<input type="hidden" name="StartTime" value="<pbs:!#StartTime#>">
<input type="hidden" name="Genre" value="<pbs:!#Genre#>">
```

The two JavaScripts in the example form are to manage the selections.

Options

Type="Times"	Extracts a list of <option> fields, which is used to select the start time for the lineup. The tag is used like this: `<pbs:tv_getoptions type="Times">`
Type="TVDates"	Extracts a list of <option> fields, which is used to select the date for the lineup. The tag is used like this: `<pbs:tv_getoptions type="TVDates">`
Type="TVChannels"	Extracts a list of <option> fields, which is used to select a single channel. The tag is used like this: `<pbs:tv_getoptions type="TVChannels" Provider="OH60418" LineUp="R" Tier="2" any=1>`

The other options are there to provide defaults. Please see pbs tv_schedule for more information.

pbs tv_LineUpData

<pbs:tv_LineUpData>

Before web guests can view TV listings, they must select their geographical location and TV provider. The TV provider is the cable provider, satellite network, and so on. The template is called **gen_lineup.pbs**. This template must be accessed with **/apps/pbcs.dll/tv_lineup.**

The tag must be used like this:

```
<pbs:!tv_LineUpData useobjects=1>
```

Options

Useobjects	Enables the use of object files.

TVLineUpSelect.pbo

Used to select the provider.

<%Change%>	Stores the previous selection.
<%LineUps%>	Outputs `<option>` fields for the drop-down.

TVLineUpZipSelect.pbo

This form is used to send the zip code to the server. Please see the example code for the form.

<%Change%>	Stores the previous selection.
<%ZipCode%>	Outputs the zip code that was previously entered.

pbs tv_programinfo

<pbs:tv_programinfo>

This page is usually presented as a pop-up window from the TV Guide. Uses the template **X_ovr.pbs**, where X is the category specified in the link to the page. The example uses Tv1_ovr.pbs.

The tag is used in the following manner:

```
<pbs:!tv_programinfo useobjects=1>
```

Options

Useobjects	This option makes the tag use object files.

TvPrograminfo.pbo

<%Actors%>	The list of actors for the program.
<%Crew%>	Contains information about the creators of the program.
<%Description%>	A description of the program.
<%ProgramType%>	The name for the type of program. This name matches the fields in the legend in the main listing.
<%Showtimes%>	The show times for the program.
<%\Title%>	The name of the program.
<%TVRating%>	The age rating.

pbs tv_schedule

<pbs:tv_schedule>

After the web guest has selected her zip code and provider, she will be directed to the page that shows the TV line-up. This page uses the **xxx_ovr.pbs** based on the category selected in the form action. The example uses Tv_ovr.pbs.

The tag is used in the following way:

```
<pbs:!tv_schedule Provider="OH60418" LineUp="R" Tier="2" Interval=120 useobjects=1 width="625">
```

Options

Interval	Determines for how many minutes the TV schedule should be output.
LineupTier	Sets the default lineup.
Provider	Sets the default provider. This option is only used if the page is opened without any parameters.
Useobjects	Enables object files.
Width	Sets the width of the output.

These object files are used both for displaying the line-up for multiple channels and for a single channel line-up.

TVSchedule.pbo

Used to output the entire schedule.

<%ColCount%>	Used by the system to set the width of the columns for the different TV programs.
<%Header%>	The content of the file TVScheduleHeaderRow.pbo
<%Legend%>	The program legend.
<%ScheduleContent%>	Outputs the schedule.
<%StationNum%>	The station number.
<%Width%>	The width set in the tag.

TVScheduleHeaderRow.pbo

Used to output the header for the listing.

<%CellInfo%>	Outputs the cells with the time headers, or the name of the channel if you are viewing the listing of one channel.
<%StationNum%>	Controls the output of either the text "channel" or "time".

TVScheduleRow.pbo

Output for each row in the listing.

<%CellInfo%>	Outputs the information about each program.
<%ChannelCallSign%>	The channel call sign.
<%ChannelNum%>	The channel number.
<%ShowStart%>	The start time of the program when listing all programs for a single channel.
<%StationNum%>	The station number. It is used when linking to the page for listing all programs for a single channel.

TVScheduleRowCell.pbo

<%Category%>	The program's category.
<%CC%>	This field has a value if the program is Closed Captioned.
<%ChannelID%>	The channel ID. This information is used to link to the details page for the program.
<%ProgramID%>	The program ID. This information is used to link to the details page for the program.
<%ProgramType%>	Sets the correct code for the style sheet.

<%Rating%>	The program's rating.
<%SchedID%>	The schedule ID. This information is used to link to the details page for the program.
<%Span%>	Sets the width of the cell.
<%\Title%>	The title of the program.
<%TooShort%>	Contains a value if the cell is too short to show the program information.

Chapter 36 Movie Guide

Movie Guide is a module that allows you to show information about cinemas in your region. This is not the same as the Cinema Module, which is based on the events calendar, and in which theaters must enter the information themselves. This movie guide requires that you have an information feed from a content provider.

- Link to own reviews
- pbs mv_movieinfo
  - MVMovieInfo.pbo
  - MVMovieInfoImageItem.pbo
  - MVMovieInfoImageRow.pbo
  - MVMovieInfoReviewItem.pbo
- pbs mv_schedule
  - MVSchedule.pbo
  - MVScheduleChainFooter.pbo
  - MVScheduleChainHeader.pbo
  - MVScheduleMovieItem.pbo
  - MVScheduleTheaterFooter.pbo
  - MVScheduleTheaterHeader.pbo
- Search form for movie guide

Link to own reviews

To attach your own reviews, use the variable

```
#filmid#
```

in the keyword portion of the **Newslist** tag. This will look for stories with the corresponding film ID as a keyword. You can combine the `#filmid#` variable with other keywords if necessary.

```
<pbs:newslist category=XXX keyword="#filmid#" days=90 count=5>
```

pbs mv_movieinfo

<pbs:mv_movieinfo>

This function uses the template **X_ovr.pbs**, where X is specified in the tag linking to the page. The example uses mv1_ovr.pbs.

Options

ImageListColumns	Controls how many columns the images should span. The option is used like this: `ImageListColumns=4`
ImageListStart	Used to skip a number of images. This option is typically used when you use an extra page to display images. The option uses the first picture that the page should show as a parameter like this: `ImageListStart=15`

Objectclass	Sets the object class.
Useobjects	Enables the use of object files.

MVMovieInfo.pbo

This file is the mail file for the **Movieinfo** tag.

<%Actors%>	The name of the actors.
<%Caption%>	The caption for the first image. `<%Caption2%>` contains the caption for the second image, and so on.
<%Description%>	The description of the film.
<%Distributors%>	The name of the distributor.
<%FilmID%>	The unique ID for the film. The ID can be used in links.
<%Genres%>	The genre information.
<%HomePage%>	The URL for the film's homepage.
<%Images%>	Outputs MVMovieInfoImageRow.pbo
<%ImageCount%>	The number of images connected to the movie.
<%ImageLink%>	The URL for the first image. The field `<ImageLink2%>` contains the URL for the second image, and so on.

<%ImageRef%>	The reference code. This is used when you want to use the zoom function on a picture. It is used like this:
	`<%pbcs.dll%>/misc?url=/templates/zoom.pbs&FilmId=<%FilmId%>&Ref=<%ImageRef2%>`
	The field
	`<%ImageRef2%>`
	contains the Reference code for the second image, and so on.

<%Rating%>	The age rating.
<%ReleaseDate%>	The release date. If no value is present, this will be empty.
<%Reviews%>	A link to reviews.
<%RunningTime%>	The running time for the film.
<%\Title%>	The title of the film.
<%WideReleaseDate%>	The nationwide release date, as read from the tvmv.dbo.FilmInfo table.

MVMovieInfoImageItem.pbo

<%Caption%>	The caption for the image.

<%FilmId%>	The ID for the film. It is used in the zoom link.
<%ImageLink%>	The URL for the image.
<%ImageRef%>	The reference code for the image. It is used in the zoom link like this: `<%pbcs.dll%>/misc?url=/templates/zoom.pbs&FilmId=<%FilmId%>&Ref=<%ImageRef%>`

MVMovieInfoImageRow.pbo

This file is output by the field

```
<%images%>
```

in MVMovieInfo.pbo. The file is output for each row of images. It will be output for as many rows as needed.

<%Col1%>	Outputs the image for the first column.You also have `<%Col2%>, <%Col3%>` and so on up to the number of columns you specified in the tag.

MVMovieInfoReviewItem.pbo

This field is output for each review.

<%Body%>	The body of the review
<%Description%>	A description of the reviewer.
<%FilmID%>	The film ID.
<%\Title%>	The title for the review.

pbs mv_schedule

<pbs:mv_schedule>

The tags can be placed on all templates. The example uses **mv_ovr.pbs**.

The tag is used like this:

```
<pbs:!mv_schedule ZipCode="43660" useobjects=1 width="625">
```

Options

Radius	Sets the search radius in miles.
Timeformat	Please note that it is not possible to include the date in **Timeformat.** It outputs the time and should be used as in this example: `<pbs:!mv_schedule zipcode="9000" useobjects=1 width="625" timeformat="H:nn:ss AM/PM">`

Useobjects	Enables the use of object files.
Width	Sets the width of the output.
Zipcode	Sets the default zip code for the listing.

MVSchedule.pbo

<%ScheduleContent%>	Outputs the content.

MVScheduleChainFooter.pbo

This file has no fields.

MVScheduleChainHeader.pbo

<%ChainName%>	The name of the movie theater chain.

MVScheduleMovieItem.pbo

<%FilmID%>	The ID for the film.
<%MovieName%>	The name of the movie.
<%Rating%>	The movie's rating.
<%RunningTime%>	The running time.
<%ShowTimes%>	The show times.
<%TheaterID%>	The ID for the movie theater that is sent in the feed to the TVMV database.

MVScheduleTheaterFooter.pbo

<%Items%>	Outputs all the movie items.

MVScheduleTheaterHeader.pbo

<%Address%>	The address.
<%City%>	The city name.
<%Phone%>	The phone number.
<%State%>	The state.
<%TheaterName%>	The name of the movie theater.
<%ZipCode%>	The zip code.

Search form for movie guide

The form is set up like this:

```
<form name="moviesearchform" action="<pbs:prog>/section?Category=MV" method="POST">
Fields and buttons go here.
</form>
```

You must set the category in the action field to the category in which the movie listing will be displayed.

Fields and buttons

Date field	Used to select the date for which movie listings will be shown. Use the Getoptions tag to get a list of dates currently in the database.
	
	`<select name="Date">`
	`</select>`

Radius	`<input name="Radius" value="<pbs:!#Radius#>">`
Search button	`<input  type="button" value="Go!" onclick="javascript:DoSubmit();">`
Zipcode	Used to enter the zip code for which movie listings will be shown. The zipcode tag is used to extract a zip code the web guest has previously entered. `<input name="ZipCode" value="<pbs:!#ZipCode#>">`

The JavaScripts included in the examples above are required for the search form to work.

The form can use the following tags:

Get options	Extracts options from the database. To extract available dates, use the tag like this: `<pbs:tv_getoptions type="MVDates" dateformat="dddd, mmmm d">` The **Dateformat** option is used to set the format of the dates shown in the drop-down.
Radius	Extracts a previously entered search radius. The tag is used like this: `<pbs:!#Radius#>`
Zipcode	Extracts a previously entered zip code. The tag is used like this: `<pbs:!#ZipCode#>`

Chapter 37 Other Tags

This chapter describes miscellaneous tags that are used with SAXOTECH Online.

- Internet Explorer functions
- pbs calendar
  - Calendar.pbo
  - CalendarDateItem.pbo
  - CalendarWeekItem.pbo
- pbs category
- pbs categorydescription
- pbs categoryid
- pbs catselect
- pbs datecalc
- pbs datechoice
- pbs delimiter
  - Delimiter.pbo
- pbs dingbats
  - DingBats.pbo
- pbs frontpagearchive
  - FPArchive.pbo
  - FPArchiveItem.pbo

- pbs getcookie
- pbs getheaderinfo
  - HeaderInfo.pbo
- pbs GetKeywords
  - KeywordList.pbo
- pbs lastupdated
- pbs link
- pbs metarefresh
- pbs parvariables name="site"
- pbs PbcsRequestRandomA
- pbs PbcsRequestRandomB
- pbs profileid
- pbs profilename
  - ProfileName.pbo
- pbs prog
- pbs query
  - CustomQuery_queryname.pbo
  - CustomQuery_querynameitem.pbo
- pbs queryparam
- pbs random
- pbs script
- pbs sessioninfo
  - ArticleExtLogin.pbo
- pbs SetHeaderInfo
- pbs showdate
- pbs statuslog
- pbs UniquePageID
- Pop-up calendar
- sessioninfo with static cache files

Internet Explorer functions

The following Internet Explorer functions can be used with SAXOTECH Online.

Make us your homepage

This JavaScript triggers the **Set homepage** function in the browser. The text "somelink" should link to a page containing information about setting the default homepage for browsers that do not support this function.

```
<a href="http://www.website.com/somelink/"
onClick="if (IE4)
{
this.style.behavior='url(#default#homepage)';
this.setHomePage('http://www.website.com/');
return false
}"
TITLE="Make us your homepage">Make us your homepage</a>
```

Add to favorites

```
<a href="javascript:window.external.AddFavorite('http://www.website.com', 'Website.com!')"
target="_top">Add us to your favorites!</a>
```

pbs calendar

<pbs:calendar>

This tag outputs a calendar that you can use to browse your content. This tag can be used on all templates.

Options

Eventdates	Determines whether only dates with events should be active. Enter `eventdates=1` to enable this option.
Futuredates	Determines whether the calendar should show future dates. Enter `futuredates=0` to disable this option.
Objectclass	Sets the object class for the calendar.
Pastdates	Determines whether the calendar should show past dates. Enter `pastdates=0` to disable this option.
Weekstart	Sets the first day of the week. 1 is Sunday, 2 is Monday, and so on.

URL Options

CalDate	Determines which month the calendar should show.

Calendar.pbo

<%CalChangeScript%>	Outputs a script that can be used with the previous and next month fields to browse the calendar. It can be used like this:
	`<a href="javascript:calChange('<%PrevCalDate%>')">Previous</a>`
	`<a href="javascript:calChange('<%NextCalDate%>')">Next</a>`

<%\MonthName%>	The name of the current month.
<%NextCalDate%>	The next month in YYYYMM format.
<%NextMonthName%>	The name of the next month.
<%NextShortMonthName%>	The name of the next month in short form.
<%PrevCalDate%>	The previous month in YYYYMM format.
<%PrevMonthName%>	The name of the previous month.
<%PrevShortMonthName%>	The name of the previous month in short form.
<%ShortMonthName%>	The name of the current month in short form.
<%WeekItems%>	Lists the week items.
<%\Year%>	The year.

CalendarDateItem.pbo

<%Active%>	This field has the value 1 if the date is active. It is speculated that this variable will have a value of 1 if there is an event that occurs on the date currently being output (need confirmation from Development).

<%\Date%>	The date.
<%FirstInWeek%>	This field has the value 1 if the day is the first in the week.
<%ISODate%>	The date in ISO format. It can be used like this: `<a href="events?Category=EVENTS&Date=<%ISODate%>&CalDate=<%ISODate%>">`
<%LastInWeek%>	This field has the value 1 if the day is the last in the week.
<%NotInMonth%>	This field has the value 1 if the day is not in the current month.
<%Today%>	This field has the value 1 if it is today's date.

CalendarWeekItem.pbo

<%CurrentWeek%>	This field has the value 1 if it is the current week.
<%DateItems%>	Lists all the days in the week.
<%WeekNumber%>	Contains the week number.

pbs category

<pbs:category>

This tag outputs the name of the main category for the section category currently being viewed.

Options

Category	Makes the tag output the story's category
Main	Makes the tag output the name of the main category when viewing a subcategory.
Section	Makes the tag output the section category instead of the story's category. The section category is inherited from the section front on which the story is displayed. If the section category is blank, this option has no effect.

By using the tag as follows:

```
<pbs:category section main>
```

you will get the name of the maincategory for the sectioncat you are viewing.

pbs categorydescription

<pbs:categorydescription>

This tag outputs the category description or information. The description is stored in the database. Without options, the category description field is output.

Options

Category	Makes the tag output the category.
Info	By default, this option is not used. Enter information in the tag as an option to output the info field instead instead of the description field.

pbs categoryid

<pbs:categoryid>

This tag outputs the name of the category ID currently being viewed. This function can be used to include an image based on the category. It can be used like this:

```
<img src="/image/header_<pbs:categoryid>">
```

Because the include tags are parsed first, the tag **cannot** be used in the include tag like this:

```
<!--#include virtual="/includes/<pbs:categoryid>.inc"-->
```

Options

Main	Makes the tag output the ID of the main category when a subcategory is being viewed.
Section	Makes the tag output the section category instead of the story's category. The section category is inherited from the section front on which the story is displayed. If section category is blank, this option has no effect.

pbs catselect

<pbs:catselect>

Enables you to output text based on the category being viewed. This tag can be used in all templates.

The tag might appear as follows:

```
<pbs:catselect selection="frontpage,news" values="#3265CB,#003399" default="#3265CB">
```

Options

Default	Sets the default value.
Selection	Used for the list of categories.
Values	Used for the list of values for each category in the selection option.

pbs datecalc

<pbs:datecalc>

The tag for date calculation is called `<pbs:datecalc>`. It allows you to do date calculations for search forms and other fields.

Without any options, the tag will output the active publication date.

Options

Dateformat	This tag supports the SAXOTECH Online Date format for output. Date format is documented in the section Formatting dates.
Days	Sets the offset from today's date in days. By default, this is not set. The days can be set to any value. This is added onto the publication date. You can also set days=-4 to subtract four days from the publication date.
Firstofmonth	Sets the date to the first day of the month.
Firstofweek	Sets the date to the first day of the week.
Lastofmonth	Sets the date to the last day of the month.
Lastofweek	Sets the date to the last day of the week.
Months	Sets the offset from today's date in months. By default, this is not used. The months can be set to any value. This is added onto the publication date. You can also set months=-4 to subtract four months from the publication date.
Now	Outputs today's date.
URL	Makes the tag use the date in the URL as the start date for the calculation.

Weekstart	Determines whether Sunday or Monday is the first day of the week. Setting this option to 0 makes Sunday the first day of the week. Setting it to 1 makes Monday the first day of the week. 0 is the default. You can set Weekstart to be any number from 0 to 6.

The tag is most often used in search tags to set date ranges. To make a form that searches this month's stories, use:

```
<pbs:datecalc firstofmonth>
```

in the **from date** field, and

```
<pbs:datecalc lastofmonth>
```

in the **to date** field.

To search the last 7 days of stories, use:

```
<pbs:datecalc days=-7>
```

in the **from date** field, and

```
<pbs:datecalc now>
```

in the **to date** field.

To search only the previous month's stories, use:

```
<pbs:datecalc firstofmonth days=-2 firstofmonth>
```

in the **from date** field, and

```
<pbs:datecalc firstofmonth days=-2 lastofmonth>
```

in the **to date** field.

pbs datechoice

<pbs:datechoice>

This tag is used to create a list of date options for a drop-down menu. It always uses todayÕs date.

Options

Dateformat	Sets the format for the date that is displayed.
Days	Sets the number of days for which the system should create options.

The tag

```
<pbs:datechoice days=3 dateformat="mm/dd/yyyy">
```

will output this:

```
<option value="20030806" selected>Today</option>
```

```
<option value="20030805">08/05/2003</option>
```

```
<option value="20030804">08/04/2003</option>
```

pbs delimiter

<pbs:delimiter>

This function can be used everywhere to output a delimiter.

Delimiter.pbo

This file has no fields.

pbs dingbats

<pbs:dingbats>

This function is used to output special characters. It can be used everywhere.

Options

Char	Determines which character should be shown. You use the option like this: `<pbs:dingbats char="n">`

DingBats.pbo

<%Character%>	This field contains the value from the char option.

An example of the object file output would be:

```
<img src="/images/dingbats/<%Character%>.gif" border=0>
```

pbs frontpagearchive

<pbs:frontpagearchive>

This tag makes links to the front pages for the last week. This tag can be placed on all templates. If you are going to use this option, your **Newslist** tags on the front page must contain the option `ado=1`.

The tag might appear as follows:

```
<pbs:frontpagearchive days=14  sort=desc dateformat="dddd, mmmm dd">
```

Options

Dateformat	Used to set the date format. For more information, please see Formatting dates.
Days	Determines the number of days for which the tag should output links. By default, links are output for 6 days. This option can be set from 1 to 6 days.

Sort	Used to sort the list of dates. By default, days are sorted in ascending order. Enter `sort=desc` to sort the days in descending order.
Weekstart	Used to show the front pages from yesterday to the start of the week. By default, this option is not used. To use this option, set the option to the first day of the week.

This tag will use object files if they exist.

FPArchive.pbo

<%FPArchiveItems%>	Contains all links.

FPArchiveItem.pbo

<%\Date%>	The date in the formatted format.
<%ISODate%>	The date in ISO format.
<%LastItem%>	Contains a space for the last item. This field can be used for extra spacing after the last record.

pbs getcookie

<pbs:getcookie>

This tag is used to read a cookie. Use a script to set a cookie. This tag can be used on all templates.

Options

Name	This option specifies the name of the cookie you want to read. By default, this option is not set. Enter `name="XXX"` where XXX is the name of the cookie.

pbs getheaderinfo

<pbs:getheaderinfo>

This tag is used to read HTTP headers.

Options

ExtendedHeaders	Used to specify extra headers to be read. By default, this option is not used. Any headers not included in the HeaderInfo.pbo variable list must first be defined here. For example, entering: `<pbs:getheaderinfo ExtendedHeaders="Accept-Encoding">` will allow access to a variable named <%Accept-Encoding%> in the HeaderInfo.pbo.

HeaderInfo.pbo

These fields are derived form the message headers defined in the standard HTTP protocol. Any headers not included in this list must first be pre-defined in the pbs getheaderinfo tag using the ExtendedHeaders parameter.

<%Accept%>	Used to specify certain media types, which are acceptable for the response. Example: `Accept=text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/pla`
<%Authorization%>	Consists of credentials containing the authentication information of the client for the realm of the resource b
<%CacheControl%>	The cache control settings sent by the browser. Example: `CacheControl=no-cache`
<%Connection%>	Specifies options that are desired for the particular connection. This must not be communicated by proxies `Connection=keep-alive`
<%Content%>	The entity-body that is sent. For example, names and values from a form. Example: `Content=SearchTitle=Cars&SearchBody=Volvo&SearchBodyTitle=Brand&SearchRegionID`
<%ContentEncoding%>	Specifies which data format transformations (called content (en)codings), such as compression mechanism assigns preferences to them.
<%ContentType%>	Specifies the Internet media type of the entity-body that is sent or would have been sent if requested. Often the character encoding. Example: `ContentType=application/x-www-form-urlencodedHost=localhost`
<%ContentVersion%>	Defines the version tag associated with a rendition of an evolving entity. Combined with the **Derived-From** simultaneously on the creation of a work as an iterative process. The field should be used to allow evolution should not be used to indicate derived works or renditions in different representations. It may also be used entity's version with that of the current resource.
<%DerivedFrom%>	When an edited object is resubmitted (for example, using PUT), this field gives the value of the Version . Th two concurrent modifications by different parties will not be lost, and to use established version manageme
<%Host%>	Specifies the Internet host of the resource being requested. Example: `Host=localhost`
<%InternalPathInfo%>	Contains the path to the page being shown. Example: `InternalPathInfo=/frontpage`
<%InternalScriptName%>	Contains the path to the SAXOTECH Online dll. Example: `InternalScriptName=/apps/pbcs.dll`
<%Method%>	The method for accessing the page. Example: `Method=GET`

<%PathInfo%>	Additional path information, as given by the client. This consists of the trailing part of the URL, after the scri Example: `PathInfo=/frontpage`
<%PathTranslated%>	The value of **PathInfo,** but with any virtual path name expanded into a directory specification. Example: `PathTranslated=c:\inetpub\wwwroot\frontpage`
<%ProtocolVersion%>	Gives the version of the information retrieval protocol relating to this request. Example: `ProtocolVersion=HTTP/1.1`
<%Referer%>	Used by a client to specify the address (URL) of the resource from which the Request-URL was obtained. T `Referer=http://localhost/apps/pbcs.dll/frontpage?nocache=1`
<%RemoteAddr%>	The IP address of the client or agent of the client (for example, gateway or firewall) that sent the request. E `RemoteAddr=127.0.0.1`
<%RemoteHost%>	The host name of the client or agent of the client (for example, gateway or firewall) that sent the request. If should set **RemoteAddr** to leave this unset.
<%ScriptName%>	A virtual path to the script being executed, used for self-referencing URLs. Example: `ScriptName=/apps/pbcs.dll`
<%Title%>	The title of the document. Not part of the document. The syntax is the same as for the **Subject** field in RFC **Subject** field in that the title is defined by the author of a document, but the **Subject** field is defined by the
<%URL%>	Gives the base portion of the URL. Example: `URL=/apps/pbcs.dll`
<%UserAgent%>	Contains information about the user agent (client) originating the request. Example: `UserAgent=Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.0.7) Gecko/2`

pbs GetKeywords

<pbs:GetKeywords>

This tag extracts keywords from stories. It is typically used on a search page to populate a drop-down. This tag can be used on all templates.

Options

Categories	Use this option to extract keywords from several categories. Example: `categories="NEWS%,SPORTS%"`

Category	Sets the category from which the keywords will be extracted. Example: `category="NEWS"`
Objectclass	Sets the object class for the object file.
Options	When `options=1` is used, the object file is not used and the tag will output `<option>` tags for a `<select>` list.

The tag is used to output options for the **keyword** field in a search form:

```
<select name="Keyword" size="1">
<option value=" " SELECTED>Select Topic</option>
<pbs:getkeywords category="NOTICES" options=1>
</select>
```

KeywordList.pbo

<%Keyword%>	Outputs the keyword.

pbs lastupdated

<pbs:lastupdated>

Used to show the date on which a story was last updated. This tag can be used on all templates.

The tag is used like this:

```
<pbs:showdate format="ddd dd.mm">
```

Options

Format	Sets the date format. The legal values are described in the section Formatting dates.
Kategori	Determines the category for which the date will be shown.Kategori has the following options for use: • kategori="%"* Shows the last updated date for all categories. • kategori="NEWS"* Shows the date stories in the NEWS category were last updated. • kategori="NEWS01,NEWS02"* Shows the date stories in the NEWS01 or NEWS02 categories were last updated. • kategori="NEWS%"* Shows the date stories in all of the NEWS categories were last updated.
Lower	Makes the system output the text in lowercase.
Norwproper	Makes the system output the text with the first letter in the first word in uppercase.
Proper	Makes the system output the text with the first letter in each word in uppercase.
Upper	Makes the system output the text in uppercase.

pbs link

<pbs:link>

```
<pbs:link article inherit template=printart text="Printer-friendly version">
<pbs:link>
<pbs:link inherit ondate=0 href text=img:>&MaxW=580">
```

pbs metarefresh

<pbs:metarefresh>

This tag is put in the <head></head> section to enable pages to auto refresh. This tag can only be used on front pages, section fronts, and stories. The {{<pbs:metarefresh> }}tag has only one option, refresh time. The refresh time is set in the **web.siteconfig** table. You can set separate refresh times for the front page, section front, and story. The default refresh time is 10 minutes.

pbs parvariables name="site"

pbs:#parvariables# name="site"

It is currently not possible to use all story parameters as global variables. This tag can be used to retrieve parameters:

```
pbs:#parvariables# name="site"
```

Where name can be set to the following values:

site	The site indicated in the publicus.ini.
date	The date in the URL. If there is no date, the current date will be used.
category	The category in the URL.
storyid	The story ID in the URL. This is only available on the /article action.
publishdate	The current site publish date.
ref	The story or photo gallery AR/PH.
theme	The theme from the URL.
keywords	The keywords from the URL.
profile	The profile ID from the URL.
sectioncat	The section category from the URL.
lightid	The light ID from the URL.

pbs PbcsRequestRandomA

<pbs:#@PbcsRequestRandomA#>

This tag is processed when using static cache files. It is a random number between 0 and 1000000.
To enable this tag for post processing make the following publicus.ini setting:

```
[CACHE]
EnablePostProcessRandomNumbers=1
```

pbs PbcsRequestRandomB

<pbs:#@PbcsRequestRandomB#>

This tag is processed when using static cache files. It is a random number between 0 and 1000000.
To enable this tag for post processing make the following publicus.ini setting:

```
[CACHE]
EnablePostProcessRandomNumbers=1
```

pbs profileid

<pbs:profileid>

This option gives you the current ID for the profile. This tag works in the same way as pbs categoryid.

pbs profilename

<pbs:profilename>

This tag extracts the profile name on a story.

Options

ID	Specifies which ID the tag should show. By default, it shows the current profile. Enter `id=x` to show the name for profile x.
Objectclass	Sets the object class.

ProfileName.pbo

<%IsCurrent%>	This field has the value 1 if the profile is the current profile.
<%ProfileID%>	The profile ID.
<%ProfileName%>	The profile name.

pbs prog

<pbs:prog>

To output the path to the SAXOTECH Online dll, you have two functions. These functions should be used instead of writing the path directly in the document.

In pbs files, you can use the function `<pbs:prog>` to output the path to the SAXOTECH Online dll.

In pbo files you can use the function `<%pbcs.dll%>` to output the path to the SAXOTECH Online dll.

pbs query

<pbs:query>

The tag is used to execute a stored procedure that you have written and output the content as part of the page. The tag is written as follows:

```
<pbs:query name="Queryname">
```

It will look for the stored procedure in the database. You must must have direct access to the database so you can execute queries and create stored procedures. Custom queries can only use stored procedures. In the stored procedure you can reference:

- Stored procedures
- Tables
- Functions

All references to functions, stored procedures, and tables require full database and owner qualifiers, for example, **custom.dbo.mystoredprocedure**. You can also make references outside the custom database. When doing this you also need full qualifiers, such as **web.dbo.brukere**. In addition, all selects should use the **(nolock)** hint to ignore locks that might slow down the query.

The tag has the following options:

Objectclass	This option sets the object class for the object files. This option is not required.

Name	Name sets the name of the stored procedure. Name is required. The name must contain only the following characters: • Letters from A to Z (both upper case and lower case) • Numbers from 0 to 9 • Underscore (_)

Parameters	This option is used for sending parameters to the query you have written. This option is not required. The parameters are sent from the tag must match the parameters the query requires. Otherwise, the query will fail. You can use @ to specify a specific parameter name. All parameters need values; stored procedures may specify defaults.The parameters should be passed using the syntax:
	`"parametername"="parametervalue"`
	So the query tag might appear as follows:
	`<pbs:query Name="foo" parameters= "@date=6/20/2008,@count=<%count%>,@startdate=<%start%>">`

In this example, we use the parameters option to pass three parameters to the query in the order specified:

```
<pbs:query Name="foo" parameters="@date=6/20/2008,@count=<%count%>,@startdate=<%start%>">
```

In this example, two of the parameters are sent as blank:

```
<pbs:query Name="foo" parameters="@date=6/20/2008,,">
```

In this example, the query is inside a macro accepting fields. The count parameter is skipped, so **@start=** is used to specify the last parameter:

```
<pbs:query Name="foo" parameters="@date=6/20/2008,@startdate=<%start%>">
```

If you send in more or less variables than the system expects, the system will expose fully qualified SQL server messages on the results page. Always use parameter names. This makes a longer string but makes no difference in performance. Sending all parameters makes the tag easier to read and follow later.

The tag can be placed directly in a template. The tag must then be run cached. If you need to run the tag with recache intervals or you need to pass in variables, the tag must be placed inside a macro or an <pbs:include>.

For more information and examples of custom queries, please see the SAXOTECH Online Custom Queries manual.

CustomQuery_queryname.pbo

Replace **queryname** with the name of the query you are running.

<%items%>	This field outputs all items.

CustomQuery_querynameitem.pbo

Replace **queryname** with the name of the query you are running. All column names become fields in the object file. In addition, the following fields are available:

<%counter%>	This field contains the counter value for the record shown.
<%islastitem%>	This field has the value 1 on the last record.
<%odd%>	The field has the value 1 if the record number is odd.
<%even%>	The field has the value 1 if the record number is even.

pbs queryparam

<pbs:queryparam>

This tag outputs query parameters on the page. A query parameter is the part of the URL that comes after the ? character. This function is usually used to put values from the link in forms for further use.

Options

Param	Used to select the parameter that will be output.

The following code will output the category ID for the category being viewed:

```
<pbs:queryparam param=category>
```

pbs random

<pbs:random>

This tag is used to create a variable with a random number. This number can be used in other functions.

Options

ID	Sets the name for the global variable. If the ID is set to xxx, the value will be called `<%@RandomValue_xxx%>`
Max	Sets the maximum value.
Min	Sets the minimum value.

This example shows how you can output different tags based on the random value.

The tag might appear as follows:

```
<pbs:include virtual="/includes/ads.inc" cachetime=2>
```

The Ads.inc might appear as follows:

```
<pbs:random min=1 max=3 id=adselect>
<pbs:macro name=adselect>
```

The AdSelect.pbo might appear as follows:

```
[%{@RandomValue_adselect=1}
<pbs:clasifieds......jobs>
%]
[%{@RandomValue_adselect=2}
<pbs:clasifieds......houses>
%]
[%{@RandomValue_adselect=3}
<pbs:clasifieds......autos>
%]
```

pbs script

<pbs:script>

The script tag is used to execute scripts written in other languages from within the SAXOTECH Online publishing platform. This function uses the Windows Scripting Host. By default, VBScript and JavaScript are supported. The SAXOTECH Online publishing platform supports any language supported by the Windows Scripting Host, including Perl, PHP, Python, .NET, and so on.

Saxotech strongly recommends that prior to using the script tag you receive additional training and read the SAXOTECH Online Scripting Technical User Manual.

All scripts must be placed in the **scripts** folder in the **templates** folder. The typical location is **/templates/scripts.**

The basic tag looks like this:

```
<pbs:script name="scriptname" language="Vbscript">
```

Options

Name	Sets the name of the script.

Language	The language sets the script language used. By default, only VBScript and JavaScript are installed on the web server. If you want to use additional script languages, you must install support for the languages on the server before they can be used. The language used in the file must match the language specified in the tag. **There are errors in the following list of examples. For instance, when using PHP5, you should NOT specify a language; instead, append the extension '.php5' to the end of the script name specified in the tag.** Examples:
	`<pbs:script name="scriptname" language="VBScript">`
	The system will look for the file **scriptname.vbs.**
	`<pbs:script name="scriptname" language="Jscript">`
	The system will look for the file **scriptname.js.**
	`<pbs:script name="scriptname" language="PerlScript">`
	The system will look for the file **scriptname.pl.**
	`<pbs:script name="scriptname" language="PHPScript">`
	The system will look for the file **scriptname.php5.**
	`<pbs:script name="scriptname" language="ActivePHP">`
	The system will look for the file **scriptname.phs *or*** scriptname.php.*
	`<pbs:script name="scriptname" language="Python">`
	The system will look for the file **scriptname.py.**
	`<pbs:script name="scriptname" language="RubyScript">`
	The system will look for the file **scriptname.rb.** All other languages use the extension `.wsf` for the script file.

Cache time	The tag does not support the cache time parameter. You must place the tag inside a macro or an include file.

Variables can be defined on the fly in the tag using the same method as the `<pbs:macro>` tag.

The $S modifier can be used on variables in any object file that is in the SAXOTECH Online Tags Manual. This modifier is used to call a script. Note that you cannot use memory banks when using the $S modifier.

The SAXOTECH Online object is used to communicate with Online from inside the script. For the complete list of properties and methods available for the SAXOTECH Online object, please see the SAXOTECH Online Scripting Technical User Manual.

pbs sessioninfo

<pbs:sessioninfo>

It is possible to control access to stories and photo galleries. Use Access Control to give or limit access to certain stories. For more information on Access Control, please see the *SAXOTECH Online User Manual*. To set up Access Control in the ini file please see AccessControl. Access Control uses the template **gen_art_extlogin.pbs**.

This functionality uses either the `<pbs:sessioninfo>` tag or the `<pbs:#sessioninfo#>` tag. You can one of these tags on a login page to retrieve information from the session object. Always use `<pbs:#sessioninfo#>` when using full page caching to avoid registered user information being cached in the cache file and presented to other users.

The tag takes no parameters.

Use the `Create a SessionInfo.pbo` file, located in the objects folder and containing these variables:

```
<%LoginName%><%FirstName%><%LastName%><%EmailAdr%>
```

ArticleExtLogin.pbo

<%AccessControlIdentifier%>	Indicates the flow ID that is associated with a story in the edit interface. If there are no flows associated with the story, then this variable will be empty. This variable is used in conjunction with Access Control.
<%CurrentFlowStatusID%>	Represents the current status of the story within a flow using an ID. This variable is used in conjunction with Access Control.
<%CurrentFlowStatusText%>	Represents the current status of the story within a flow using text. This variable is used in conjunction with Access Control.
<%SingleSaleID%>	Represents the single Sale ID associated to a story. It is possible to associate a single sale id to a story in the editing interface. This is used for Access Control.
<%StandardText%>	This field contains the text to output. For example, *You must be a subscriber to access this story. To buy a subscription...*

pbs SetHeaderInfo

<pbs:SetHeaderInfo>

This tag is used to change values of the HTTP header fields available through pbs getheaderinfo, either as the standard HTTP header fields or fields defined through the pbs getheaderinfo parameter.

The

```
<pbs:SetHeaderInfo>
```

tag is used by adding parameters, of the form Name=value. This will add a new header or overwrite an existing one of the same name.

Example 1	`<pbs:SetHeaderInfo HeaderName="HeaderData" ... >`
Example 2	`<pbs:SetHeaderInfo Accept-Encoding="Yes" ... >`

Example 3	<pbs:setheaderinfo remotehost=newXPhost> <pbs:getheaderinfo> If your HeaderInfo.pbo file is like the following: <%remoteaddr%> <%remotehost%> Then where the output was previously Gandalf192.168.172.1It will now be newXPhost192.168.172.1 ||

pbs showdate

<pbs:showdate>

This function allows you to output a date. This tag can be used on all templates.

The tag is used like this:

```
<pbs:showdate format="ddd dd.mm" offset=17>
```

Options

Format	Sets the date format. The legal values are described in the section Formatting dates.
Lower	Makes the system output the text in lowercase.
Norwproper	Makes the system output the text with the first letter in the first word in uppercase.
Offset	Determines how many days the date is offset from today's date.
Proper	Makes the system output the text with the first letter in each word in uppercase.
Upper	Makes the system output the text in uppercase.

pbs statuslog

<pbs:#statuslog#>

The tag <pbs:#statuslog#> is a debugging function that outputs the build time of the page. This tag should be included commented out at the bottom of your templates so you can easily check the build time for the page.

pbs UniquePageID

<pbs:#@UniquePageID#>

Performs a lookup in the localdata.ini file to find a mapping for the following actions:

- /frontpage (default value frontpage)
- /CCE (default value CCE+Module)
- /section (default value category)

To enable this tag for post processing make the following publicus.ini setting:

```
[CACHE]
EnablePostProcessRandomNumbers=1
```

Pop-up calendar

The pop-up calendar is used when you want the web guest to be able to pick a date from a calendar. This function is typically used for the events listing.

The following JavaScript is needed to open the calendar:

```
<script language="javascript">
<!--
function choosedate () {
 window.open('/g/kalender_eng.html?
formname.fieldname&nobackdate'
,'','menubar=0,titlebar=0,width=268,height=236');
 document.searchform.Interval.options[document.searchform.Interval.options.length-1].selected =
true;
}
function ResetDate () {
 if (document.searchform.Interval.selectedIndex !=
(document.searchform.Interval.options.length-1))
 {
  document.searchform.dateselected.value = "";
 }
}
-->
</script>
```

Options

Formname.fieldname	Used to select the field in which the date will be placed. Both the form name and the field name must be specified.
Limit	Used to set the limit for the dates that can be picked. For example: `&limit=20010101-20011231` The option can be used with the pbs datecalc tag.
Nobackdate	Used to block access to dates before today.
Submit	Determines that the form will be automatically submitted when a date is chosen.
Submit=	Determines which button in the form should be selected when the date is chosen.

sessioninfo with static cache files

<pbs:#@sessioninfo#>

This tag is processed when using static cache files. It calls the sessioninfo function and uses object class 999

The `<pbs:#@sessioninfo#>` will be processed in before dispatch thus global variables defined during the process of this tag will be available during the processing of all tags. This tag will not be processed for the following actions:
`/STATUS, /VER, /ALIVE, /EXEC`

<pbs:#@sessioninfoarticle#>

This tag is processed when using static cache files. It calls the sessioninfo function and uses object class 998. As the name of the tag indicates this tag is only processed for `/ARTICLE` action.

The `<pbs:#@sessioninfoarticle#>` will be processed in before dispatch thus global variables defined during the process of this tag will be available during the processing of all tags. This tag will not be processed for the following actions:
`/STATUS, /VER, /ALIVE, /EXEC`

Like the existing `<pbs:#sessioninfo#>` tag the processed content will only be added during the post processing of static cache files if the following is set in the publicus.ini:

```
[CACHE]
EnablePostProcessSessionInfo=1
```

Chapter 38 Community Site

Community Site (previously known as Publicus Light) is the Community Publishing Module, which allows your customers to maintain their own sites as a part of your website. Your customers have access to the same functions that you have.

A Community Site can use the following features:

- Stories
- Links

All other tags will pull information from the main site. A Community Site can also pull information from a main siteÕs database. This allows the site to show stories from a category or stories tagged with a keyword or a theme.

- Other tags and community site
- pbs article and community site
- pbs buildmenu
    - MenuGraphicItem.pbo
    - MenuTextItem.pbo
- pbs lightid
- pbs links on community site
- pbs sectionfront on community site
- Community site setup
- Community site templates

Other tags and community site

All other tags will pull information from the main site's database.

pbs article and community site

<pbs:article>

The article tag does not support object files on Community Sites.

pbs buildmenu

<pbs:buildmenu>

Used to build Community Site menus, this tag builds a sorted menu based on the value in the **Sort** field in the category register.

MenuGraphicItem.pbo

<%link%>	The link to the section front.
<%Image%>	Used to show an image based on the category. It is used like this:{{<a href="<%link%>"><img src="<%Image%>" border=0> }}The system looks for an image called **menu_<category>.gif** in the folder: **/light/<LightID>/graphics**

Example

```
<a href="<%link%>"><img src="<%Image%>" border=0><br></a><br>
```

MenuTextItem.pbo

<%link%>	The link to the section front.
<%Categoryname%>	The name of the category.

Example

```
<a href="<%link%>" class="menu" style="color: #000000;">
<%Categoryname%></a><br>
```

pbs lightid

<pbs:lightid>

This tag has no options.

pbs links on community site

<pbs:links>

The links tag supports object files on Community Sites.

pbs sectionfront on community site

<pbs:sectionfront>

Builds a section front for the Community Site. The tag follows normal behavior.

Community site setup

To enable Community Site, the option

```
light=1
```

must be set in your Publicus.ini in the MODULES section.

Create Users

To create a Community Site user, you must log on to the editing interface as a system administrator.

When you create the new user, you must use the **Light ID** field. This ID field is a numerical value from 001 to 999. At least one user for each light ID must be given system administrator rights. These rights apply only to the Community Site, and not the entire website.

Instructions for creating users can be found in the *SAXOTECH Online User Manual.*

Create categories

After you have created a user, the user must log on and create categories. These categories can have any names. The main category must be named FRONTPAGE. These categories are separate from the main siteÕs categories.

Alias

To give the users easier access to the Community Sites hosted by you, you should create an alias that redirects users to the main page. If you have created a Community Site customer with ID 001 and the category FRONTPAGE for the front page, the alias must point to:

`/apps/pbcs.dll/section?Category=FRONTPAGE&LID=001`

Community site templates

The templates which are used by Community Site are:

gen_ovr.pbs
gen_art.pbs

You can also create templates for each category. These templates follow default SAXOTECH Online behavior. Community Sites do not

use the **frontpage.pbs** template.

The templates for each Community Site reside in this folder:

```
/light/<customerid>
```

Chapter 39 Xsendmail

XSendMail is a component that handles forms on web pages. It takes information in the form and translates it into an e-mail, which is sent to the specified address.

To use XSendMail, the XSendMail.ini file must be present in the folder above your application folder.

- XSendMail.pbo
- XSendMail Form
- XSendMailHTML.pbo
- XSendMailHTMLItem.pbo
- XSendMailItem.pbo
- XSendMail setup

XSendMail.pbo

<%Copyright%>	Copyright message
<%ImpMessage%>	The "Do not reply to this email message" message.
<%Items%>	Returns the values from the XSendMailItem.pbo.
<%Title%>	Only used for XSendMailHTML.pbo. The result of the title field from the Send Mail form.

XSendMail Form

The form for XSendMail has the following structure:

```
<form method="post" action="/apps/XSendMail.dll">
Fields go here.
Submit button.
</form>
```

Templates

The form can be put on all pages. The example template is called **XSendMail_ovr.pbs**.

Fields

You can create any fields you like, as long as they do not have the same names as any of the hidden fields.

Hidden fields

These fields are put in the form as hidden fields.

Body	This is a text field that is put in the e-mail. By default, this field has no value. To use this field, enter a value similar to the following: `<input type="hidden" name="Body" value="This is a message">`

File	Allows web guest to select a local file to be sent with the form. `<input type="file" name="fileupload">`
From address	Hides the sender address. By default, the setting in the server-wide XSendMail.ini is used. Enter the following to hide the sender's address: `<input type="hidden" name="from" value="server@website.com">`
From name	Hides the sender name. By default, the setting in the server-wide XSendMail.ini is used. Enter the following to hide the sender's name: `<input type="hidden" name="fromname" value="The Webserver">`
Hide recipient	Disables the output of the recipientÕs e-mail address in the e-mail sent to the person filling in the form. By default, the e-mail address is shown. Enter the following to suppress the recipient's address: `<input type="hidden" name="hiderecipient" value="1">`
Language	Sets the language for the e-mail and system messages. It is typically used like this: `<input type="hidden" name="lang" value="XXX">` By default, the language is Norwegian. Other languages available use the following values: • **NOR** Norwegian • **SWE** Swedish • **ENG** English • **DAN** Danish
No HTML	Used to disable HTML in the e-mail. By default, e-mail is output as HTML. To send e-mail as plain text use the following: `<input type="hidden" name="nohtml" value="1">`
No receipt	Determines whether a receipt is sent to the sender. By default, a receipt is sent. Enter the following if no receipt should be sent to the sender: `<input type="hidden" name="noreceipt" value="1">`
Objectclass	Sets the object class. By default, object files are not used. Objectclass is set as follows: `<input type="hidden" name="objectclass" value="23">`
Required	Determines which fields are required. The web guest will view missing fields listed on an error page. It is important to use field names that make sense to the web guest.By default no fields are required. Enter the following to set required fields: `<input type="hidden" name="REQUIRED" value="Name,E-mail">`

Return_link	This option is used to set the return link on the error page. It is used as follows: `<input type="hidden" name="Return_link" value="javascript:history.go(-1)">` By default the value is javascript:history.go(-1). You can use any URL for the return address.
Return_text	Sets the text for the return link. By default the return text is "Tilbake til skjema." This can be changed to any text string.
Redirect_URL	Sets the URL to which the web guest is directed after the form is filled in and submitted correctly. This option is required.By default, this option has no value. Enter a value similar to the following: `<input type="hidden" name="REDIRECT_URL" value="/misc/thankyou.htm">`
Subject	Sets the subject for the e-mail. By default, the field has no value. Enter the following to create a subject: `<input type="hidden" name="SUBJECT" value="Your Subject Text">`
Title	Sets the title for the form. The field is used like this: `<input type="hidden" name="TITLE" value="A test form">`
To	Sets the e-mail address to which the form is sent. This option must be set using the following tag: `<input type="hidden" name="TO" value="someone@website.com">` If you want the e-mail to be sent to multiple recipients, you can specify multiple e-mail addresses in a semicolon separated list.

Validatekey	The validatekey validates against spam. The option is activated using the following:
	
	`<input type="hidden" name="validatekey" value="<pbs:!validatetag>">`
	
	Note that two settings must be set in the XSendmail.ini in the GENERAL section:
	
	`KeyVerificationEnabled=1`
	`KeyVerificationLifeTime=X`
	
	where X is the number of minutes for the lifetime of the key.

XSendMailHTML.pbo

This file is used for the HTML version of the e-mail. It has the same fields as XSendMail.pbo.

XSendMailHTMLItem.pbo

This file is used for the HTML version of the email. It has the same fields as XSendMailItem.pbo.

XSendMailItem.pbo

<%Name%>	The name of the field.
<%Value%>	The value of the field.

XSendMail setup

Xsendmail requires its own ini file called **default_XSendMail.ini**. To make a site specific version, follow the standard SAXOTECH Online logic.

GENERAL section

The XSendMail.ini GENERAL section has the following settings:

Captcha	The user's solution to the captcha. Protects forums using a captcha using the <pbs:captcha> tag. To enable this setting, enter `Captcha  = 1` The default is 0. The form submitted must contain the CaptchaID value.
BypassVerification	When enabled, the three verification steps for XSendMail are disabled for testing purposes. To enable, enter `BypassVerification=1` The default is disabled.
From	Sets the default email address from which messages are sent.
Fromname	Sets the default sender name.
KeyVerificationEnabled	Used to turn key verification on and off. The default is 1 (enabled).
KeyVerificationLifetime	The key lifetime in minutes. (0...N). The default is 30.
Mailserver	The fully qualified address is the address of the SMTP server that will send the email. You can also use the IP address instead of the address.
ReferrerVerificationEnabled	Used to turn referrer verification on and off. The default is 1.
ReferrerVerificationMethod	Used to determine which approach is used when trying to match the allowed referrers to the actual referrers. Note that all referrer matching is case sensitive. For example: `ReferrerVerificationMethod=[REGEXP*|INSTR|HEAD|EXACT]` Where: • REGEXP matches by searching for the allowed referrer as a regular expression • INSTR matches by searching for the allowed referrer in the referrer string • HEAD matches the allowed referrer to the head/start of the actual referrer • EXACT performs an exact match
ReferrerVerificationSection	The name of the section in the ini file where the allowed referrers are listed. It defaults to REFERRERS, which is the list used by the old verification routine. `ReferrerVerificationSection=[REFERRERS*|<newsection>]`
ReferrerVerificationShowDetails	Used to make XSendMail report details on why verification failed, including the actual referrer and the valid referrers loaded from the ini file. The default is 0 (disabled).

UseNewVerificationScheme	Set to 1 to enable the new verification scheme. The default is 0. The verification scheme performs the following three checks: • Checks to make sure the referrer simply matches the domain from Publicus.ini. Referrer is the Request.Referrer. • If it does not directly match the domain-setting, the same thing is checked for all values in the REFERRERS section in the XSendMail.ini. Then the systems checks whether the allowed host has been added to the ini-file.

Example:

```
[GENERAL]
Mailserver=fully.qualified.address
From=webserver@somesite.com
Fromname=The Webserver
UseNewVerificationScheme=0
KeyVerificationEnabled=1
KeyVerificationLifeTime=30
ReferrerVerificationEnabled=1
ReferrerVerificationShowDetails=0
ReferrerVerificationMethod=REGEXP
ReferrerVerificationSection=REFERRERS
```

REFERRERS section

This section is used to list the actual referrers. For example:

```
[REFERRERS]
www.mydomain.com
http://www.anotherdomain.com
```

Chapter 40 URL Rewrite

URL Rewrite is a function that recodes SAXOTECH Online URLs into a shorter, more user-friendly format. This benefits both you and your web guests. Users can easily read and remember the new URLs. The new URL format is also easier for search engines to read, because the URLs appear to be static. This will lead to increased indexing of your site on search engines.

A standard SAXOTECH Online URL looks like this:

`http://www.somesite.com/apps/pbcs.dll/section?Category=NEWS`

With URL Rewrite enabled, it will look like this:

`http://www.somesite.com/section/NEWS`

- New URLs
- Potential problems
- URL Parameters
- URL Rewrite setup
- Using URL Rewrite

New URLs

When the function has been enabled, the following new URLs are available for use:

```
/ = /apps/pbcs.dll/frontpage
/section = /apps/pbcs.dll/section
/classifieds = /apps/pbcs.dll/classifieds
/misc = /apps/pbcs.dll/misc
/article = /apps/pbcs.dll/article
/search =  /apps/pbcs.dll/search
```

If you have any folders with these names on the web server, you will **not** be able to access them when URL Rewrite is enabled.

Potential problems

If you have created files or folders in the root folder of your website with names matching the SAXOTECH Online action, these pages will no longer be available when URL Rewrite is enabled.

If you have coded your links as relative links, your site will break.

Incorrect link:

```
<a href="section?category=NEWS">News</a>
```

Correct links:

```
<a href="/apps/pbcs.dll/section?category=NEWS">News</a>
<a href="<pbs:prog>/section?category=NEWS">News</a>
```

All links on your site must be fixed so they conform to this format before the urlrewrite.dll can be installed for your web servers.

URL Parameters

You can control what the page displays by putting parameters in the URL. This is done by adding **&params=** to the end of the URL.

The `&params` setting has the following options:

Edate	This option is used to specify the end date. The default is 99999999. Any date in ISO format can be used.
Eindex	Sets the stop letter.
Sdate	This option is used to specify the start date. The default is 19800101. Any date in ISO format can be used.
Search	Controls if the tag should be executed. By default, the search is executed. By putting `Search=0` in the URL, the tag will not be executed. This is useful if you would like to build a search page so the user must click on a link or fill in a form to get a search result.
Sindex	Sets the start letter.

URL Rewrite setup

The setup for URL Rewrite consists of three parts:

Webserver setup

The dll urlrewrite.dll must be installed as a filter for the website. This can be done either for each server, or for each site. The dll should be placed in a separate folder if it applies to the entire server, or in the websiteÕs application folder if it applies to one website only. This installation must be done by the people managing your web server. This is not done at site level.

default_rewrites.ini

To use URL Rewrite, the file **default_rewrites.ini** must be present in the folder above your application folder. The ini file must be called default_rewrites.ini. To make a site specific version, replace "default" with the application pool name for the website. The file can look like this:

```
[SETTINGS]
Debug=0
RootRewrite=0
LogIncomingURL=0
Forum2NewsRewrite=0
ForumRewrite=1
MiscRewrite=0
SearchRewrite=0
ClassifiedsRewrite=0
EventRewrite=0
GuideRewrite=0
GalleryRewrite=0

[REWRITES]
^/exec\?(.*)=/apps/pbcs.dll/exec?$1
^/([a-zA-Z0-9\-\_]+)/([0-9]+)-([0-9]+)-(.*?).html$=/apps/pbcs.dll/article?AID=/$2/$1/$3
```

ClassifiedsRewrite	Enter
	`ClassifiedsRewrite=1`
	to enable rewrite of classified page URLs. The default is
	`ClassifiedsRewrite=0.`

Debug	Enter
	`Debug=1`
	to enables debug mode. The default is
	`Debug=0`

EventRewrite	Enter
	`EventRewrite=1`
	to enable rewrite of event URLs. The default is
	`EventRewrite=0.`

Forum2NewsRewrite	When
	`Forum2NewsRewrite=1`
	`/forum` in the URL is changed to `/news`.
	The default is
	`Forum2NewsRewrite=0`

ForumRewrite	Enter
	`ForumRewrite=0`
	to disable the rewrite of forum URLs. The default is
	`ForumRewrite=1`

GalleryRewrite	Enter
	`GalleryRewrite=1`
	to enable rewrite of image gallery URLs. The default is
	`GalleryRewrite=0.`

GuideRewrite	Enter
	`GuideRewrite=1`
	to enable rewrite of event URLs. The default is
	`GuideRewrite=0.`

LogIncomingURL	Enter
	`LogIncomingURL=0`
	to disable the logging of the URL before and after the rewrite. The default is
	`LogIncomingURL=1.`
	In order for **LogIncomingURL** to work, the **Debug** setting must also be enabled.

MiscRewrite	Disables the rewriting of the /misc URL. The default is enabled. When it is turned on, files placed in the **misc** folder cannot be accessed directly. With
	`MiscRewrite=0`
	the URL looks like this:`[http://www.website.com/apps/pbcs.dll/misc?template=zoom.pbs`
	With
	`MiscRewrite=1`
	the URL looks like this:`http://www.website.com/misc/etc`

[Rewrites]	Used to add frequently used shortcuts. For example, you could have `http://www.site.com/tennessee%20titans` point to `/apps/pbcs/dll/section?Category=SPORTS04`. To do this you would need the following line in the [REWRITES] section: `^/Tennessee\stitans=/apps/pbcs.dll/section?Category=SPORTS04`

RootRewrite	Enables the rewriting of the front page (root) URL to /. The default is disabled. With
	`RootRewrite=0`
	the URL looks like this:`http://www.website.com/apps/pbcs.dll/frontpage`
	With
	`RootRewrite=1`
	the URL looks like this:`http://www.website.com/`

SearchRewrite	Enter
	`SearchRewrite=1`
	to enable rewrite of search result URLs. The default is
	`SearchRewrite=0.`

[Settings]	Used as a placeholder for the settings. Each of these settings enables URL rewrite for separate sections within a server (default) or site (XX).

Publicus.ini

In order for the system to write the new function, URLrewrite must be enabled in the GENERAL section of the website's Publicus.ini. The option is written like this:

```
URLrewrite=1
```

Using URL Rewrite

The format for a URL when URL Rewrite is enabled is different than when it is disabled. With URL Rewrite enabled, the story URL will look like this:

```
http://servername/article/20041019/REPOSITORY/110180004/1001/NEWSLIST02
```

With URL Rewrite disabled, the story URL looks like the examples below:

```
http://servername/apps/pbcs.dll/article?AID=/20041019/REPOSITORY/110180004/1001/NEWSLIST02
http:
//servername/apps/pbcs.dll/artikkel?Dato=20041019&Kategori=REPOSITORY&Lopenr=110180004&Profile=100
http:
//servername/apps/pbcs.dll/article?Date=20041019&Category=REPOSITORY&id=110180004&Profile=1001&Sec
```

The new URL format is shorter and easier to read. / is used as a separator instead of &. This will allow Google, Yahoo and other popular search engines to index all your pages.

When URL Rewrite is enabled, the following tags will write the new URL form automatically:

```
<pbs:ft>
<pbs:newslist>
<pbs:agent>
```

You can create your own short URLs for use in menus and other places. You an use either URL format.

Chapter 41 Go.dll

This is a method for tracking clicks.

Use	`<a href="/apps/go.dll?http://www.publicus.com/">Go! Go!</a>`
Logging	The clicks are logged in the folder **/adlogs.** The log file is called ads *<isodate>_<number>.log*

guide_GuideAdItem

GuideAdItem.pbo

The GuideAdItem makes a list of the ads for a given guide.

<%Active%>	Is the item active?
<%ActiveImageCounter%>	Number of images in active items.
<%AdDescription%>	Description of the ad item.
<%AdID%>	ID of the ad item.
<%AdText%>	Text of the ad item.
<%AdTitle%>	Title of the ad item.
<%EndDate%>	Last appearance date in the guide for an ad item.
<%GuideAdShowFullSize%>	
<%GuideID%>	ID of the guide for the ad item.
<%ImageCounter%>	Number of images in an ad item.
<%StartDate%>	First appearance date in the guide for an ad item.
<%UnsavedListIndex%>	Always -1

guide_GuideMainItem

GuideMainItem.pbo

Output for each main category in the system.

<%GuideMainCat%>	The title of the main category
<%GuideMainCatLink%>	The link to the list of all the entries in the main category.
<%GuideSubCatItems%>	Outputs all the subcategories.